N° d'ordre : 2362

THESE

présentée

pour obtenir

LE TITRE DE DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE

École doctorale : Informatique et Télécommunications

Spécialité : Réseaux et Télécommunications

Par M. Mahamadou ISSOUFOU TIADO

Titre de la thèse Modèles et Mécanismes multiniveaux pour les réseaux sans fil

Soutenue le 13/07/2006 devant le jury composé de :

M.	Michel DIAZ	Président
MM.	André-Luc BEYLOT	Directeur de thèse
	Monique BECKER	Rapporteur
	Khaldoun AL AGHA	Rapporteur
	Riadh DHAOU	Membre
	Guillaume URVOY	Membre
	Fabrice ARNAL	Invité

Thèse de Doctorat

Présentée pour obtenir le grade de

Docteur de l'Institut National Polytechnique de Toulouse

Spécialité : ***Réseaux et Télécommunications***

Présentée par

Mahamadou ISSOUFOU TIADO

MODELES ET MECANISMES MULTINIVEAUX POUR LES RESEAUX SANS FIL.

Thèse soutenue le 13 Juillet 2006 devant le Jury composé de :

Michel DIAZ	*Président*
Monique BECKER	*Rapporteurs*
Khaldoun ALAGHA	
Riadh DHAOU	*Examinateurs*
Guillaume URVOY	
Fabrice ARNAL	*Invité*
André-Luc Beylot	*Directeur de Thèse*

Table des matières

Liste des figures

Liste des tableaux

Résumé

Les réseaux ad–hoc sont une particularité de réseaux informatiques, constitués de nœuds mobiles qui utilisent un mode de communication sans infrastructure et des liaisons radios. Chaque nœud mobile communique dans son rayon de portée d'émission/réception, et est totalement autonome quant à son déplacement, son fonctionnement et sa participation à l'acheminement des informations du réseau.

L'utilisation des réseaux ad–hoc présente de nouveaux enjeux de part les problèmes cruciaux qu'ils posent, notamment les problèmes liés au support de communication qui est hertzien et donc de qualité variable dans l'espace et dans le temps. Les enjeux s'étendent également à la couche d'accès au support (par exemple Wi-Fi), à la couche réseau (en particulier aux algorithmes de routage) et à la couche transport (le comportement de TCP est sensible aux variations de délai). L'utilisation des liaisons radios introduit des différences notoires et de nouvelles problématiques par rapport aux communications filaires telles que la limitation physique ou réglementaire de la capacité disponible pour l'accès radio, la qualité fluctuante des liens radios (influence des obstacles, du mouvement, des interférences, …), la position des points d'accès inconnue à l'avance et variable dans le temps … De part ces limitations qui font que les réseaux sans fil sont moins performants que les réseaux câblés, les protocoles du modèle en couches du réseau câblé ne peuvent être transférés dans l'environnement sans fil sans adaptation. Un des enjeux en terme de recherche qui est apparu, est d'optimiser le fonctionnement des réseaux ad–hoc à travers l'utilisation de techniques innovantes qui permettent d'améliorer leurs performances. Les techniques multi-niveaux appelées "cross–layer" sont ainsi apparues pour faciliter le partage d'information entre les couches du modèle OSI et s'appliquent à tous les protocoles de divers niveaux, tant qu'il existe des interactions pour lesquelles les performances globales du système peuvent être améliorées.

Cette thèse traite des multiples aspects de la mise en place de modèles et mécanismes cross–layer dans le réseau ad–hoc. Elle permet de régler les premiers problèmes liés à l'introduction d'un nouveau mode de communication des protocoles de la pile du modèle OSI. En effet, la proposition de la méthode de conception RCL (Reverse Cross–Layer) de modèles cross–layer permet de conserver les acquis de cette architecture, à savoir, l'aisance de la conception modulaire, la définition systématique des interactions entre les composants, la poursuite des objectifs à long terme quant à l'utilisation des réseaux. La temporisation persistante proposée au niveau des protocoles fiables de la couche transport lorsque le canal sans fil à état variable est mauvais, vise à améliorer la latence, le débit de transmission et le taux de tentatives infructueuses coûteuses en terme de consommation d'énergie qui sont des caractéristiques du traditionnel back–off exponentiel. Les simulations effectuées dans l'environnement ns-2 ont permis d'évaluer les gains de performance obtenus par usage de la temporisation persistante.

De même, cette thèse consacre la proposition de mécanismes cross–layer complémentaires tel que le mécanisme cross–layer d'évaluation continue de l'état du canal en fonction de l'activité ambiante et des protocoles de routage utilisés. La standardisation des informations cross–layer fournies par les protocoles de routage a été proposée pour répondre à la nécessité de fonctionnement des modèles cross–layer indépendamment de la nature pro-active ou réactive des protocoles de routage. La proposition du mécanisme cross–layer de temporisation unifiée vise à optimiser la consommation d'énergie dans le cas de duplication de la temporisation d'attente d'envoi à des échelles de temps différentes, qui survient lorsque la couche transport fiable est associée à un protocole de routage ré–actif.

Introduction

Les révolutions informatique et télécommunications opérées ces dernières années convergent vers l'omniprésence des services du réseau Internet réalisée grâce à des connexions sans fil, dans des environnements fixes (à domicile et sur les lieux de travail) et dans des environnements mobiles (automobiles, avions, trains, bateaux, …), ce qui a pour effet de favoriser la croissance de l'utilisation d'ordinateurs portables et de présenter de nouveaux enjeux en terme de recherche, d'investissement, de revenu.

L'environnement sans fil fondé sur l'utilisation des liaisons radios est caractérisé par deux modes de communication à savoir le mode avec infrastructure où la communication entre deux points ou nœuds quelconques du réseau passe nécessairement par une station de base et les réseaux mobiles sans infrastructure (appelés réseaux ad–hoc) où chaque point ou nœud du réseau peut communiquer directement avec ses voisins, sans station de base intermédiaire.

Les réseaux ad–hoc sont une catégorie de réseaux sans fil caractérisés entre autres par une topologie dynamique, une bande passante limitée, des contraintes de consommation d'énergie. Le lien sans fil a la caractéristique essentielle de varier en fonction du temps et de l'espace. Il est aussi caractérisé par une mémoire à courte échelle due au multi–trajet pouvant causer une rafale d'erreurs qui occasionne l'impossibilité de transmettre correctement les paquets sur le lien. Le changement de l'état du lien sans fil de "bon" à "mauvais" et vice–versa peut intervenir de façon asynchrone pendant des laps de temps très courts. En plus de la variation du canal à petite échelle, il y a une variation à grande échelle qui implique le conditionnement de l'état moyen du canal par la position de l'utilisateur et le niveau des interférences possibles [SHA03].

De façon générale, les réseaux sans fils sont moins performants que les réseaux filaires. C'est pourquoi les protocoles du modèle en couches du réseau câblé ne peuvent être transférés dans l'environnement sans fil sans adaptation. Il est donc important de situer l'utilisation des réseaux ad–hoc dans le cadre de techniques innovantes destinées à améliorer leurs performances.

L'architecture OSI en sept couches pour l'interconnexion des systèmes ouverts destinée aux réseaux sur laquelle repose l'architecture actuelle d'Internet reste une base incontournable de conception des réseaux sans fil. A priori, les protocoles de ce modèle en couches destinés aux réseaux filaires sont conçus de façon indépendante les uns des autres, en ce sens que dans ce modèle, la fin de l'exécution d'un protocole de niveau inférieur qui a consommé ses données au niveau du nœud destinataire par exemple, ne doit pas influencer l'exécution des protocoles de niveau supérieur.

Le passage des systèmes reposant sur les réseaux filaires vers les environnements sans fil et la différence notoire qui existe entre ces deux environnements ont conduit à l'émergence des systèmes multi-niveaux ou Cross–Layer pour répondre aux besoins de l'amélioration des performances qui s'imposent au bénéfice des systèmes sans fil dont par exemple les réseaux ad–hoc. Le concept de « Cross–Layer » introduit une technique d'adaptation des protocoles au contexte sans fil à travers le partage d'information entre les couches. Cette technique permet d'obtenir des gains de performance divers comme démontré dans plusieurs études, ce qui justifie pleinement l'importance des modèles réseaux « Cross–Layer ».

La technique « Cross–Layer » s'applique à tous les protocoles de divers niveaux du modèle en couches, tant qu'il existe des interactions pour lesquelles les performances globales du système peuvent être améliorées. Un exemple simple relatif au multimédia illustre l'importance de cette conception novatrice. La gestion de ressources, l'adaptation et les stratégies de protection disponibles dans les couches inférieures de la pile du modèle OSI (PHY, MAC, Réseau, Transport) sont optimisées sans considérer explicitement les caractéristiques spécifiques des applications multimédias par exemple, les algorithmes de flux et de compression ne considèrent pas les mécanismes fournis par les couches inférieures pour la protection d'erreur, l'ordonnancement, la gestion de ressource, ainsi de suite.

L'architecture du modèle en couches a été éprouvée et a joué un rôle important dans la durée de vie et dans l'expansion du réseau Internet. Dans cette architecture, La couche physique traite des signaux et fournit un service de communication de bits. La couche liaison de données fournit l'abstraction du lien et la possibilité de transmettre et de recevoir une trame sur le lien. La couche réseau introduit le concept de route ou chemin, qui est une séquence de liens. La couche transport fournit un tunnel de bout en bout, qui peut être fiable ou pas, en fonction du protocole utilisé. Les trois dernières couches sont moins bien définies et ont été regroupées dans l'architecture TCP/IP au sein de la couche application. L'interaction entre les couches est contrôlée et conduite à travers les entêtes des protocoles. En plus, l'architecture implique une notion d'équivalence de protocoles qui agissent en médiateur entre les couches correspondantes sur des nœuds différents. A partir de cette architecture, les concepteurs peuvent se concentrer sur la conception des protocoles d'une couche donnée avec l'assurance que le système global fonctionnera raisonnablement bien lors de l'intégration.

L'architecture permet de diviser un système en des composants modulaires lors de sa conception et définit systématiquement les interactions entre les composants. Cette modularité fournit les abstractions nécessaires au programmeur pour comprendre le système global. Elle accélère à la fois la conception et l'implantation du système en permettant de tirer profit de l'effort parallèle. Les concepteurs peuvent focaliser leurs efforts sur un sous–système particulier avec l'assurance que l'ensemble du système sera opérationnel. Une bonne architecture favorise une prolifération rapide et la longévité des systèmes. Les modules individuels peuvent être mis à jour, sans nécessité de "reconception" du système global qui risque d'étouffer le développement futur et la longévité du système.

Il y a lieu de reconsidérer le principe de la conception cross–layer dont le but est d'améliorer les performances des réseaux, pour permettre de conserver les avantages offerts par l'architecture. En effet, une conception cross–layer débridée peut conduire à un système en forme de « spaghetti » [KAW05], de telle sorte que toute modification d'amélioration devient difficile du fait qu'une nouvelle modification interagira avec une multitudes d'autres déjà existantes et qu'au lieu de modifier une seule couche, c'est l'ensemble du système qui risque d'être modifié.

Cette conciliation des gains obtenus grâce aux mécanismes cross–layer avec les avantages offerts par l'architecture peut se faire à notre sens à travers une bonne méthode de conception des systèmes cross–layer. Pour davantage d'efficacité de la méthode de conception, nous introduisons la notion de modèles conceptuels cross–layer. Ces modèles sont une représentation abstraite des interactions et permettent d'expliciter toutes leurs implications possibles. Ils peuvent prendre diverses formes et permettent d'organiser efficacement la convertibilité systèmes filaires – systèmes sans fil tout en faisant ressortir le travail de conversion à réaliser. C'est pourquoi après avoir présenté l'état de l'art en matière de conception et de modèles cross–layer, nous proposons au chapitre II la méthode de conception de modèles cross–layer que nous appelons méthode RCL (reverse Cross–layer) qui permet d'isoler sous le nom d'Action Atomique Cross–layer (AACL) les interactions cross–layer potentielles entre les protocoles et les services de différentes couches. Ces interactions seront par la suite organisées dans divers modèles conceptuels pour aboutir à l'énonciation des modifications à apporter à chaque protocole ou service d'une couche donnée.

La méthode RCL est appliquée à un exemple de couche de protocoles et services. L'utilisation possible d'une partie des ces AACL par le protocole TCP nous a permis de proposer le principe de temporisation persistante au niveau de la couche transport pour prendre en compte la variation de l'état du canal sans fil. Au chapitre III, nous procédons à une évaluation par simulation de la proposition d'application du principe de temporisation persistante au protocole TCP, avec une étude comparative de cette temporisation par rapport à la temporisation traditionnelle utilisée. De même, nous appliquons au chapitre IV la temporisation persistante au protocole SCTP et procédons à une comparaison des résultats de simulation entre les deux modèles de temporisation, puis entre les deux protocoles fiables de la couche transport. Le chapitre V présente les propositions que nous faisons de l'apport du routage à la temporisation persistante, de l'introduction du principe de temporisation unifiée entre la couche transport et la couche réseau (au niveau des protocoles de routage) et la proposition de standardisation des interactions fournies par les protocoles de routage pour une meilleure transparence de leur fonctionnement vis à vis des autres couches. La méthode RCL a été éprouvée dans les phases de mise en place et de mise à jour des modèles conceptuels cross–layer. Son utilisation traduit le caractère structurée de l'implantation des modèles cross–layer dans la logique de la conservation des acquis de l'architecture. Les modèles mathématiques et les résultats de simulations nous ont permis de confirmer l'efficacité de l'utilisation de la temporisation persistante au niveau des protocoles fiables de la couche transport, pour répondre à la variation dynamique de l'état du canal sans fil.

Chapitre I. Etat de l'art

I.1. Le concept d'adaptation

Une caractéristique essentielle des réseaux ad–hoc est que, tout comme les autres réseaux sans fil, ils sont soumis aux aléas d'un comportement dynamique. Les conditions du canal de transmission sont imprévisibles et variables dans le temps (dégradation de la qualité du lien , mobilité, multi-trajet, interférences, …).

En réponse à cette caractéristique importante des réseaux ad–hoc, la technique cross–layer permet aux protocoles du modèle en pile de s'adapter continuellement à l'environnement sans fil, au fur et à mesure de son évolution. C'est pourquoi pour introduire cette notion, nous avons choisi d'expliciter l'usage du principe d'adaptation fait par divers protocoles.

L'adaptation peut être définie comme étant l'aptitude que les protocoles et les applications ont de se conformer aux variations des conditions du réseau. Elle se traduit de façon concrète par divers mécanismes tels que le changement du débit de transmission, l'usage d'un mécanisme de transmission robuste et fiable, l'utilisation efficace de l'énergie, du spectre radio pour les réseaux sans fil, … Les nœuds sans fil par exemple ajustent leur puissance de transmission et leur débit pour mieux acheminer le trafic de paquets.

L'adaptation se fait moyennant une contrepartie : le changement du débit d'émission lorsque le réseau le nécessite améliore l'efficacité des protocoles mais se traduit par une oscillation des débits des données.

Par exemple, le protocole TCP de la couche transport s'adapte aux conditions du réseau au moyen de mécanismes de contrôle de congestion, de contrôle de flux, tout comme l'utilisation du mécanisme de retransmission. L'idée derrière le contrôle de congestion dans les réseaux filaires consiste à avoir recours au principe d'adaptation en ajustant les débits de transmission des sources émettrices en fonction de la charge du réseau. Les pertes de paquets détectées par une source TCP sont interprétées comme un état de congestion du réseau. La source réagit en diminuant le débit de transmission pour éviter d'injecter du trafic supplémentaire dans le réseau congestionné. Ce mécanisme de contrôle de congestion par déduction à partir des pertes de paquets supposées a été adapté à l'environnement sans fil grâce à des interactions cross–layer en utilisant un principe simple pour notifier l'état de congestion à la source TCP. Un autre aspect de l'adaptation utilisée par le protocole TCP concerne le contrôle de flux mis en œuvre pour éviter de déborder la mémoire du récepteur.

Le document [ZHE03] fait référence à l'usage du principe d'adaptation au travers de la conception cross–layer par prise en compte du comportement dynamique des réseaux sans fil. Cet exemple traite également de l'écoulement du trafic multimédia qui représente un trafic sensible mettant en valeur les optimisations cross–layer. Les auteurs définissent trois niveaux d'adaptation à la variation du canal sans fil : l'adaptation de la transmission des données des applications qui fait référence à la capacité de l'application d'ajuster son comportement en s'auto-configurant lorsqu'elle peut identifier les variations du réseau et du canal sous-jacents ; l'adaptation de la transmission des segments de la couche transport permettant au protocole de différencier différents modèles de perte de paquets (pertes dues à la congestion et pertes dues aux erreurs canal) pour invoquer le contrôle de congestion et la régularisation du débit de transmission ; l'adaptation de la couche réseau et des couches liaisons pour que ces couches distinguent différents flux d'information et appliquent le traitement adéquat en fonction du niveau de priorité requis.

L'objectif visé à travers la mise en place des interactions cross–layer entre les couches est de permettre à diverses couches d'utiliser l'information détenue par d'autres couches ou tout simplement fournie par un service d'une couche donnée, afin d'adapter leur comportement en fonction de la variation de cette information.

I.2. Problématique de la conception cross–layer

La conception cross–layer peut être perçue comme la mise en place de mécanismes destinés à optimiser le fonctionnement global d'un système par la sélection d'une stratégie conjointe à travers diverses couches du modèle OSI. L'optimisation cross–layer dépend de l'usage qui en est fait et de l'orientation choisie. Divers couches de ce modèle fonctionnent sur des unités de trafic différentes et prennent comme entrée, différents types d'information. La couche physique par exemple est concernée par les symboles et dépend fortement des caractéristiques du canal, tandis que la couche application, dans l'exemple du multimédia, est concernée par la sémantique et les dépendances entre flux, et dépend fortement du contenu multimédia. Les algorithmes et les protocoles de diverses couches du modèle OSI sont souvent conçues pour optimiser chaque couche donnée et souvent avec des objectifs différents.

Le mécanisme de signaux déclencheurs entre couches a été intensivement utilisé à la fois dans les réseaux filaires et sans fil. Ces signaux sont prédéfinis pour notifier certains événements tels que l'échec de transmission de données entre protocoles. La conception complète cross–layer introduit un principe assez large d'interdépendances entre couches pour optimiser les performances du réseau. Dans les modèles cross–layer, les protocoles utilisent des informations provenant de tous les niveaux de la pile de protocole pour adapter leur comportement en conséquence.

Les auteurs du document [SCH05] proposent une classification des solutions cross–layer en fonction du sens des interactions entre les couches. Ils distinguent notamment l'approche de haut en bas dans laquelle les protocoles des couches supérieures optimisent leur paramètres et les stratégies aux couches inférieures. Cette approche a été implantée dans beaucoup de systèmes existants, dans lesquels l'application dicte les paramètres et stratégies de la couche MAC, tandis que la couche MAC choisit le schéma de modulation optimale de la couche PHY. Ils proposent également une approche de bas en haut dans laquelle les couches inférieures essaient d'isoler les couches supérieures des pertes et des variations de la bande passante. Tandis que dans l'approche centrée sur l'application, la couche application

optimise les paramètres des couches inférieures, un paramètre à la fois, dans une orientation de haut en bas ou de bas en haut (à partir de la couche PHY), suivant ses besoins ; dans l'approche centrée sur la couche MAC, la couche application envoie les informations sur son trafic et ses besoins à la couche MAC, qui décide quels flux/messages de la couche application doivent être transmis et à quel niveau de QoS. La couche MAC décide aussi des paramètres de la couche PHY sur la base de l'information canal disponible.

I.3. Les propositions de systèmes Cross–Layer

I.3.1. Systèmes multi couches

Divers modèles portant sur plusieurs couches du modèle OSI reposent sur des interactions cross–layer entre différentes couches. C'est l'exemple du système *MobileMan* du document [CON04] ou du système CLASS de [WAN03].

Les auteurs du document [CON04] proposent le système *MobileMan* fondé sur le « full Cross–layer design » qu'ils opposent au « layer triggers » (signaux déclencheurs). L'architecture matérialisée par le système *MobileMan* comporte un système central « *Network Status* » qui fonctionne comme un répertoire d'informations collectées par tous les protocoles réseaux. Chaque protocole peut accéder au *Network Status* pour partager ses données avec d'autres protocoles, pour éviter une duplication d'efforts dans la collecte des informations internes d'état et conduire à une conception plus efficace du système. Le système *MobileMan* réalise la séparation en couche en standardisant l'accès au *Network Status* par la définition de mécanismes de lecture et écriture des données par les protocoles. Les interactions entre les protocoles et le *Network Status* sont placées derrière les échanges normaux inter-couches, permettant ainsi une optimisation sans compromettre le fonctionnement normal attendu. Toutefois, remplacer un protocole orienté état du réseau par son homologue hérité permettra à toute la pile de fonctionner correctement, au coût de fonctions d'optimisations pénalisantes. Par exemple, utiliser le protocole TCP initial implique que les optimisations cross–layer n'auront pas lieu à cette couche et que le protocole transport ne fournira aucune information au composant *Network Status*. Tout de même, toute la pile de protocoles fonctionnera normalement avec des performances dégradées. Cette architecture de référence du système *MobileMan* offre des performances dans la conception d'un réseau ad–hoc dont les optimisations cross–layer pour toutes les fonctions réseau, l'amélioration de l'adaptation locale et globale (adaptation du système à une forte variation des conditions du réseau ad–hoc pour mieux contrôler les performances du système), la pleine connaissance du contexte au niveau de toutes les couches, la réduction d'overhead dans la collection de l'information d'état du réseau.

Le document [WAN03] présente des méthodes distinctes d'implantation du « Cross–Layer » comme les méthodes « Paket Header » et « ICMP Messages ». La proposition qui est faite dans ce document concerne une méthode appelée CLASS (Cross–LAyer Signaling Shortcuts) relative à la signalisation directe entre couches non voisines. L'idée est de conserver la structure du modèle en couches tout en permettant aux messages cross–layer de se propager à travers les raccourcis locaux de signalisation hors–bande. Par exemple, permettre une communication directe entre la couche application et la couche réseau sans passer par une couche intermédiaire. Cette approche n'est pas inconnue du modèle en couche des protocoles mais n'a pas été conçue pour des fonctions de gestion génériques. Par exemple, l'entité RRM (Radio Resource Management) de la couche 3 accède directement à la couche physique dans le GSM [HAV06]. Dans le simulateur logiciel GloMoSim [GLO], une API dédiée entre la couche

réseau et la couche application a été développée. Sur la base du temps de traversée de l'interface d'une couche à l'autre et du temps de traitement au niveau de chaque couche, la mesure de la latence de propagation dans le système CLASS est meilleure. Le format de message pour la signalisation interne, comporte seulement trois champs (adresse de destination, le type d'événement, le contenu de l'événement). Les messages peuvent également être propagés dans une forme agrégée en introduisant un champ optionnel "Prochain événement". En général, un message portant sur un paramètre spécifique est généré lorsque ce paramètre change de façon significative. Les appels de fonctions sont utilisés pour positionner et extraire ces paramètres, les appels systèmes sont utilisés pour lire les messages. La conception de CLASS sert comme un cadre permettant divers implantations conformément à divers scénario d'applications.

I.3.2. Mécanismes de la couche Transport

a) Prise en compte des ajustements utilisateur

Des travaux ont été réalisés sur des protocoles spécifiques comme par exemple l'optimisation des gains de performances du TCP dans le document [RAI02]. Dans ce document, les auteurs démontrent les apports liés à l'utilisation du retour d'information des utilisateurs (couche application) et celui de la couche inférieure pour améliorer les performances de TCP. L'intuition première consiste à incorporer le retour d'information des utilisateurs dans la pile des protocoles. Par exemple, un utilisateur peut dynamiquement indiquer les priorités des applications et le système devrait en retour régler les tampons récepteurs des divers applications pour contrôler le débit de ces dernières. Il lui est également possible d'observer le comportement d'un tunnel et savoir par expérience qu'une déconnexion aura lieu dans le tunnel. Si cette information est donnée au système, pour une adaptation pro–active, plutôt qu'une réaction à la détérioration du signal ou une déconnexion, elle permettra au système de réguler les priorités des applications spécifiées par l'utilisateur. De même, un utilisateur peut lancer plusieurs applications sur une machine telles par exemple un transfert ftp et une vidéoconférence. Par défaut, le système accordera une priorité plus grande à l'application temps réel vidéoconférence plutôt qu'au ftp. En observant l'imminence d'une déconnexion, l'utilisateur peut choisir de donner une priorité plus grande au ftp. Ces exigences de priorité peuvent changer dans le temps. Les échos utilisateur peuvent être communiqués à la couche appropriée à travers un module qui prend les données entrées par l'utilisateur telle que la priorité des applications et les achemine à travers un chemin parallèle jusqu'aux couches inférieures telle que la couche TCP. Le module doit traduire les données entrées par l'utilisateur dans l'information spécifique à la couche destinatrice. Par exemple, l'information de priorité des applications peut être traduite en contrôle de la fenêtre de réception. Le retour d'information de la couche réseau est utilisé pour manipuler de façon appropriée le mécanisme de contrôle de congestion de TCP, principalement pour améliorer le débit de TCP dans les deux directions lors d'un transfert de données. Les échos cross–layer peuvent être divisés en deux catégories :

- Des couches supérieures vers les couches inférieures : ces échos peuvent être la QoS requise par l'application, les échos utilisateurs, les informations des temporisateurs de TCP pour les couches inférieures.

- Des couches inférieures vers les couches supérieures : ces échos concernent les caractéristiques du lien, les informations sur la connectivité du réseau.

b) Notification Explicite de Congestion de TCP

Le document [RAM01] présente le mécanisme cross–layer de notification explicite de congestion du protocole TCP. Dans son fonctionnement de base, le protocole TCP interprète les pertes de paquets comme une indication de congestion. Le mécanisme ECN (Explicit Congestion Notification) consiste à faire en sorte que les routeurs compatibles transport détectent la congestion avant le débordement de leur file d'attente interne, et positionnent de ce fait le bit ECN dans l'entête TCP d'un paquet reçu dans un tel état pour être acheminé. Les routeurs ne sont pas pour autant limités et peuvent toujours utiliser la perte de paquets comme indication d'une congestion. A destination, le nœud récepteur informe l'émetteur de la congestion en positionnant le bit ECN dans son paquet TCP. Lorsque l'entité TCP émettrice reçoit cette indication, il invoque son mécanisme d'évitement de congestion.

c) Notification Explicite de Perte de Paquet

Le mécanisme cross–layer de notification explicite de perte de paquet a été présenté en [BAL98]. Dans le cas des réseaux sans fil avec infrastructure où les nœuds mobiles communiquent entre eux via une station de base, un module appelé agent snoop peut être introduit au niveau de la station de base. Cet agent enregistre tous les paquets qui passent au travers des connexions TCP dans les deux sens. Il conserve la trace des segments non acquittés par le récepteur (trous/gaps) dans l'espace de séquence lorsqu'il reçoit les segments de données. Ces trous correspondent à des segments perdus sur le lien sans fil. Toutefois, il peut également s'agir d'une perte due à la congestion au niveau de la station de base. Pour éviter de marquer de façon erronée un trou de congestion comme étant dû à une perte sur le canal sans fil, l'agent snoop ajoute seulement un trou dans la liste des trous lorsque le nombre de paquets mis en attente sur l'interface d'entrée de la station de base n'atteint pas une taille maximale de sa file d'attente.

Lorsque les acquittements (ACKs), spécialement les ACKs dupliqués arrivent provenant du récepteur (indiquant ainsi la perte d'un paquet), l'agent snoop consulte sa liste de trous. Il positionne le bit ELN dans l'ACK s'il correspond à un segment de la liste avant de l'acheminer à l'expéditeur de la donnée. L'expéditeur qui reçoit un ACK possédant l'information ELN retransmet le prochain segment, mais ne prend aucune action de contrôle de congestion.

L'agent snoop détecte également la perte d'un paquet sur expiration de ses temporisateurs locaux et peut retransmettre le paquet perdu s'il l'a dans son cache.

L'utilisation de l'agent snoop sur un nœud mobile n'est pas approprié parce que les mêmes ACKs dupliqués signifient à la fois une perte par corruption et une perte par congestion. Il n'y a aucun moyen pour le mobile expéditeur de savoir si la perte du paquet est survenue sur le lien sans fil ou quelque part dans le réseau du fait d'une congestion.

d) Protocole UDP-Lite

Le pragmatisme de la suite de protocole UDP/IP a servi aux concepteurs des applications multimédia en terme de flexibilité pour créer le multimédia temps réel sur les réseaux à base d'Ethernet. Avec l'avènement des réseaux sans fil, le concept de communication temps réel entre dans un nouveau domaine. Le médium sans fil du fait de sa vulnérabilité est plus susceptible d'erreur que sa contrepartie filaire. De plus, dans le cadre des réseaux sans fil un débit binaire élevé et fiable ne peut être atteint. Néanmoins, la rapidité de déploiement des

LAN sans fil, spécialement les réseaux 802.11, a nécessité la migration des technologies de l'Ethernet filaire existantes vers le domaine sans fil, en respectant les contraintes liées à l'efficacité de l'utilisation de la bande passante et de la tolérance aux erreurs. Afin de répondre à l'augmentation du taux d'erreur, un relèvement de robustesse a été introduit dans les réseaux 802.11. Naturellement, ces protocoles robustes des couches MAC/Physique ont été déployés en conjonction avec la pile de protocoles UDP/IP testés [KHA03].

En dépit de la robustesse de la couche physique du 802.11, certaines erreurs se propagent à la couche liaison. Ces erreurs sont détectées en utilisant la séquence de test de trame (FCS : Frame Checking Sequence). La couche liaison supprime de telles trames sans distinction du nombre et de la localisation des erreurs. Un schéma suggéré en [LAR99a] et [LAR99b], essaie de tenir compte de ce mécanisme en mettant en œuvre des ajustements à la pile de protocoles notamment au niveau de la couche transport et de la couche liaison. Cette variante, appelée UDP Lite, exploite la caractéristique de tolérance aux erreurs inhérente au multimédia pour améliorer l'utilisation de la bande passante. UDP Lite repose sur le fait que les applications temps réel préfèrent souvent les paquets partiellement endommagés à la perte de paquets. Il permet de ce fait d'effectuer le checksum sur les entêtes des couches transport et application tout en ignorant le checksum sur la charge utile. Chaque message est divisé en partie sensible et non sensible. La partie sensible, qui commence à partir de l'entête de la couche transport est couverte par le checksum partiel. Le champ "longueur" de l'entête UDP est utilisé pour spécifier la taille du checksum partiel. Par conséquent, ce schéma nécessite l'abandon par la couche MAC 802.11b des retransmissions et transmet les paquets partiellement corrompus aux couches supérieures. Cette stratégie est référencée comme étant le UDP Lite tout comme en [LAR99a], la stratégie de la couche liaison est référencée comme étant PPP (Point to Point Protocol) Lite, lorsque le protocole PPP est utilisé pour le traitement de ces paquets.

I.3.3. Mécanismes de la couche Réseau

a) Généralités sur les protocoles de routage à la demande fondés sur la prise en compte de l'énergie.

Les auteurs du document [DOS02] proposent un schéma reposant sur le protocole de routage DSR (Dynamic Source Routing) [JET01] [JOH98] [DEM01] [DSR02] dans lequel chaque nœud insère la puissance requise pour atteindre le prochain saut dans le paquet RREQ de requête de route (en utilisant le contrôle de puissance) avec l'identificateur du nœud. Chaque nœud du chemin met en cache cette information et "renifle" les messages de réponse de route RREP qui ne lui sont pas destinés. En utilisant l'information mise en cache et l'information du paquet RREP, le nœud peut vérifier s'il se trouve à un chemin à moindre énergie par rapport à celui du paquet RREP. Si tel est le cas, le nœud envoie à la source une réponse gratuite de route contenant le chemin à faible consommation d'énergie.

Un protocole qui maximise la durée de vie du réseau est proposé en [MAL02], l'idée principale est que chaque nœud autre que le nœud de destination calcule le coût de son lien (utilisant la durée de vie de la batterie comme métrique) et l'ajoute à une variable indiquant le coût du chemin. Cette variable est envoyée dans l'entête des paquets RREQ. En recevant un paquet RREQ, un nœud intermédiaire démarre un temporisateur et sauvegarde le coût du chemin dans la variable *min-cost*. Si un autre paquet RREQ arrive pour la même destination et le même numéro de séquence, la valeur du *min-cost* est comparée à celle du chemin contenu dans le paquet reçu. Si le nouveau paquet a le coût de chemin le plus faible, il est diffusé, et *min-cost* prend la nouvelle valeur. Lorsque le nœud de destination reçoit le paquet RREQ, il

démarre un temporisateur et collecte tous les paquets RREQ ayant les mêmes champs source et destination. Lorsque le temporisateur expire, le nœud de destination choisit le chemin à coût d'énergie minimum et renvoie un RREP à la source.

Les auteurs du document [YU02] proposent deux protocoles : un protocole max-min similaire à celui de [MAL02], et un protocole delay-request dans lequel un nœud qui reçoit un paquet RREQ suspend sa diffusion par inondation pour une période de temps inversement proportionnelle à la durée restante de la batterie du nœud. L'intuition qui accompagne ce protocole est de permettre aux paquets RREQ qui traversent les nœuds ayant un niveau d'énergie élevé d'arriver plus tôt à destination. Ceci permet aux nœuds de découvrir les chemins qui maximisent la durée de vie du réseau.

Les protocoles de routage à la demande existants diffusent les paquets RREQ avec un seuil maximal de puissance fixé. Les auteurs du document [NI99] ont identifié 3 grands inconvénients de la diffusion par inondation notamment la redondance de la diffusion, la contention, les collisions. Ces inconvénients constituent le problème de tempête de diffusion. Un autre problème de ces protocoles de routage à la demande à prise en compte d'énergie est relatif au nombre de messages RREP. Il est connu qu'un chemin ayant un nombre élevé de nœuds faiblement distants consomme moins d'énergie qu'un chemin ayant peu de nœuds mais avec une grande portée par saut. Comme présenté ci–dessus, le schéma du document [DOS02] essaie de produire de tels chemins à optimisation d'énergie en laissant le nœud renifler les paquets RREP et envoyer les réponses gratuites de route lorsqu'il découvre que son lien est sur un chemin à moindre énergie que celui du paquet RREP. Les auteurs du document [BHU04] proposent une analyse de cette approche. L'un des problèmes évoqués est que dans les réseaux moyens ou denses, un grand nombre de réponses gratuites de route est renvoyé au nœud source. Le deuxième inconvénient de cette approche est l'overhead engendré par l'ajout de l'information sur l'énergie de chaque nœud de la route dans l'entête des paquets RREQ et dans tous les paquets de données subséquents. Dans les réseaux de grande taille, l'overhead peut être significatif du fait que le protocole essaie de minimiser l'énergie consommée en maximisant le nombre de nœuds.

b) Le protocole PCDC

Un protocole de contrôle de puissance pour les réseaux MANETs appelé PCDC est proposé en [MUQ03]. Il repose sur la transmission des messages RTS/CTS dans un canal de contrôle séparé. Le protocole PCDC permet d'effectuer des transmissions simultanées dans un même voisinage pour limiter l'interférence mais nécessite toutefois deux canaux de fréquence séparés et deux émetteurs/récepteurs, d'où son incompatibilité avec le schéma de l'IEEE 802.11. L'approche originale de ce protocole est implantée à travers le protocole CONSET dans lequel, pour assurer la compatibilité avec le standard IEEE 802.11, les auteurs utilisent un seul canal à la fois pour les paquets de données et de contrôle.

c) Le protocole CONSET

Dans le protocole CONSET [BHU04], la couche MAC affecte la performance de la couche réseau en contrôlant la puissance utilisée pour transmettre les paquets RREQ. En contrôlant la puissance de transmission des paquets RREQ, la couche MAC contrôle de manière effective l'ensemble des nœuds candidats au prochain saut. Le choix de minimiser la consommation d'énergie se traduit par la préférence d'une transmission à basse énergie, ce qui signifie aussi un petit ensemble de nœuds "prochain saut". La réduction de la taille de cet ensemble pourrait impliquer une perte de connectivité. Par conséquent, l'objectif est de fournir un mécanisme distribué par lequel un nœud peut dynamiquement traiter son ensemble de

connectivité CS (Connectivity Set) défini comme étant l'ensemble de nœuds à moindre énergie qui garantissent la connectivité du nœud dans le réseau. Le prochain saut pour la transmission de données est sélectionné à partir du CS du nœud. L'algorithme de construction du CS d'un nœud i (CS_i), avec pour objectif de produire des chemins de bout en bout à moindre énergie en maintenant simultanément la connectivité du réseau, peut être décrit comme suit :

Le CS_i doit contenir uniquement les nœuds voisins avec lesquels la communication directe nécessite moins de puissance que la communication indirecte (2 sauts) via tout autre nœud déjà existant dans le CS_i. Pour établir le CS_i, le nœud i met continuellement en cache le gain estimé du canal et l'angle d'arrivée (AOA) de chaque paquet de contrôle (RTS/CTS) qu'il reçoit, sans tenir compte de la destination du paquet. Le traitement du gain est possible du fait que les paquets de contrôle sont transmis à une puissance fixée connue et de ce fait le nœud i utilise la puissance de réception du signal pour déterminer le gain du canal. De plus, des techniques d'estimation du AOA sans un système de positionnement (tel que le GPS, ..) sont disponibles (voir [KRI97] pour les détails). Chaque nœud du CS_i est associé à un temporisateur qui expire T secondes à partir du moment où le nœud est ajouté au CS_i. Lorsque le temporisateur expire, le nœud correspondant est supprimé du CS_i. En recevant un paquet RTS/CTS provenant d'un autre nœud j, le nœud i procède comme suit : Si $j \in CS_i$, et que le nouveau gain en cours de traitement et l'AOA sont égaux à ceux qui sont stockés dans le CS_i, alors le temporisateur associé à l'entrée j du CS_i est réinitialisé et aucune autre action n'est exécutée. Sur un autre plan, si $j \notin CS_i$, ou si $j \in CS_i$ mais le nouveau gain ou l'AOA ne sont pas égaux à ceux qui sont stockés, alors le nœud i compare la puissance $P_{i,j}$ de communication directe avec le nœud j avec $P_{i,u} + P_{u,j}$ où $u \in CS_i$. Si $P_{i,j} < P_{iu} + P_{u,j}$ pout tout nœud $u \in CS_i$ alors le nœud j est ajouté au CS_i sinon il ne l'est pas. Soit $P_{conn(i)}$ la puissance minimale nécessaire au nœud i pour atteindre le nœud le plus éloigné du CS_i, si le nœud j est ajouté au CS_i et $P_{i,j} > P_{conn(i)}$, alors tout autre élément du CS_i doit être réexaminé. La raison est qu'un chemin à 2 sauts entre le nœud i et un nœud $u \in CS_i$ via le nœud j doit maintenant être plus optimal en terme de consommation de puissance que le chemin direct entre i et u. Dans ce cas, le nœud u doit être supprimé du CS_i. Toutefois, si $P_{i,j} >= P_{conn(i)}$ alors $P_{i,j} + P_{j,u} > P_{i,u}$ pour tout nœud $u \in CS_i$ et de ce fait, il n'est pas nécessaire de réexaminer le CS_i.

L'ajout ou non du nœud j au CS_i se fait en considérant uniquement les communications indirectes à deux sauts, du fait que si le chemin à deux sauts est moins optimal en terme d'énergie que le chemin direct, alors tels sont aussi les chemins à n sauts avec n >= 2 (voir [MUQ03]). Pour prendre en compte l'aspect crucial du maintien de la connectivité dans les réseaux MANETs, le document [MUQ03] démontre que si le réseau est connecté avec l'approche standard de puissance maximum, alors il peut être aussi connecté lorsque chaque nœud communique uniquement avec les nœuds de son CS.

A forte charge, il y a suffisamment d'activité RTS/CTS pour permettre de traiter le CS sans overhead hors bande. Toutefois, avec une charge légère, le canal est plus souvent libre, un schéma auxiliaire est nécessaire pour assurer le traitement précis des CS. Dans le protocole CONSET, chaque nœud diffuse un paquet "Hello" avec une puissance Pmax toutes les δ secondes avec δ une variable aléatoire uniformément distribuée dans l'intervalle [0, T]. Le caractère aléatoire est nécessaire pour éviter les collisions entre les transmissions synchronisées des "Hello". Lorsque le nœud i a traité la portée de connectivité $P_{conn(i)}$, il utilise le niveau de puissance pour diffuser ses paquets RREQ. La transmission des RREQs à la puissance $P_{conn(.)}$ engendre deux améliorations significatives. Premièrement, tout protocole de routage à nombre de sauts minimum simple tel que le protocole AODV (Adhoc On-Demand Distance Vector) [ROY00] [DAS01] [ROY99] ou le protocole DSR peuvent être utilisés pour produire des routes à énergie optimale pour augmenter le débit du réseau. De ce fait, aucune intelligence n'est nécessaire à la couche réseau et aucune information sur le lien (par exemple, la puissance) n'est

échangée ou incluse dans les paquets RREQs dans le but de trouver des routes à énergie optimale. Ceci réduit clairement la complexité et l'overhead. Deuxièmement, en considérant comment les paquets RREQ sont déversés dans le réseau, les améliorations significatives du débit et de la consommation de puissance peuvent être atteintes en limitant la diffusion de ces paquets aux nœuds se trouvant dans la portée de connectivité $P_{conn(.)}$.

I.3.4. Mécanismes des couches Liaison de données et Physique

a) Généralités sur la sélection automatique de vitesse

Le standard 802.11a offre plusieurs débits d'émission (6, 9, 12, 18, 24, 36, 48 et 54 Mb/s). Chaque débit correspond à un schéma de modulation avec ses propres compromis entre les débits des données et les distances entre stations. Le standard 802.11 et ses suppléments ne spécifient pas d'algorithme de sélection automatique de débit.

Diverses raisons ont conduit à la naissance des algorithmes de sélection automatique du débit de transmission. C'est le cas par exemple des besoins en latence faible des applications multimédias, tout comme il est important de tenir compte du fait que la probabilité de corruption d'un paquet dépend de la durée de sa transmission. En conséquence, les paquets longs doivent être potentiellement transmis à un débit plus faible que celui des paquets courts.

Dans le but de décider du meilleur débit à utiliser à chaque instant, un algorithme de contrôle nécessite des informations sur les conditions courantes du canal appelée CSI (Channel State Information). Il est difficile d'obtenir les CSI directement. C'est pourquoi plusieurs algorithmes utilisent diverses formes de retour d'information, dont par exemple les algorithmes fondés sur les statistiques tel que le niveau du débit de l'utilisateur. Le principal inconvénient du retour d'information indirect est qu'il est lent de façon inhérente et cause des coupures de communications lorsque les conditions du canal se dégradent rapidement (par exemple en cas de déplacement rapide de l'utilisateur).

Les équipements 802.11 disponibles commercialement utilisent les approches reposant sur les statistiques pour le contrôle de débit. Les algorithmes de contrôle peuvent être implantés dans le logiciel comme partie intégrante du pilote de périphérique de la carte mais aussi comme partie du chipset (ensemble de puces) qui permet de contrôler les (re)transmissions individuelles de trames.

Dans la communauté de recherche, une autre classe d'algorithmes de contrôle de débit a été étudiée. Ces algorithmes de contrôle utilisent le SNR (Signal to Noise Ratio) comme information de retour pour améliorer la sensibilité aux changements des conditions du canal. Ces algorithmes ne sont pas implantés de façon pratique dans des systèmes, seuls des résultats de simulations sont présentés.

Un moyen simple d'obtenir les informations voulues sur les conditions du canal est de maintenir des statistiques sur les données transmises telles que le taux d'erreur de trame (FER Frame Error Rate), les transmissions acquittées et le débit atteint. Ces statistiques étant liées directement au débit réel des données de niveau utilisateur, elles garantissent de façon inhérente que ce débit est maximisé sur le long terme. La simplicité et la stabilité des facteurs évoqués (statistiques) expliquent la prédominance des retours d'information fondés sur les statistiques dans les produits 802.11 actuels. Le document [HAR05] consacre une étude comparative des mécanismes de contrôle automatique de débit. On distingue trois types de contrôle de débit

fondé sur les statistiques : ceux qui reposent sur le débit, ceux qui reposent sur le taux d'erreur trame (FER) et les contrôles reposant sur les retransmissions. Les contrôles reposant sur le débit sont ceux qui utilisent le type le plus global de statistiques et sont les plus lents. Ceux qui sont fondés sur les retransmissions utilisent les statistiques les plus locales (nombre de retransmissions par trame) et cette approche est la plus rapide.

b) Contrôle de débit fondé sur le débit maximal effectif

Dans cette approche, une fraction constante de données (10%) est envoyée à la cadence donnée par les deux débits adjacents du débit courant. A la fin d'une fenêtre de décision spécifiée, les performances obtenues avec les trois débits sont déterminées en divisant le nombre de bits transmis à chaque débit par les temps cumulés de transmission. Finalement, une commutation est faite vers la valeur qui procure le plus grand débit durant la fenêtre de décision (exemple des produits 802.11a fondés sur le chipset AR5000 d'Atheros).

Pour collectionner des statistiques significatives, la fenêtre de décision doit être particulièrement large (environ 1 seconde). Par ailleurs, ceci rend l'algorithme flexible par rapport aux changements de courte durée dans la qualité du lien causés par exemple par l'évanouissement. Par ailleurs, ce mécanisme permet de prévenir les réactions immédiates aux changements de longue durée de la qualité du lien. Ce mécanisme affecte notablement les performances des flux temps réel.

c) Contrôle de débit fondé sur le FER (Frame Error Rate)

Dans ce schéma, le FER du flux de données transmis sur le lien est utilisé pour sélectionner un débit approprié. Le FER peut aisément être déterminé du fait qu'avec le 802.11, toutes les trames de données sont explicitement acquittées par l'envoi d'un ACK à l'expéditeur. De ce fait, la perte d'un ACK est une forte indication de la perte d'une trame de données. En comptant le nombre de trames ACK reçues et le nombre de trames de données envoyées pendant une assez petite fenêtre de décision, le FER peut être traité comme un ratio des deux paramètres.

Le FER peut être utilisé pour sélectionner le débit de la prochaine fenêtre comme suit [BRA01] : pour rétrograder, si le FER dépasse un seuil et que le débit courant n'est pas le débit minimal, alors le système peut commuter vers le débit immédiatement inférieur. Pour accélérer, si le FER vaut zéro (ou inférieur à un autre seuil), le système peut commuter vers débit immédiatement supérieur avec une latence de quelques trames (généralement même d'une seule). Si toutes ont été acquittées, le système commute vers le bon débit. Pour éviter que l'algorithme de contrôle ne soit sujet à des oscillations entre deux débits adjacents, le passage à un débit supérieur est interdit pendant une durée donnée après avoir rétrogradé.

L'influence de la durée de la fenêtre et des seuils mentionnés ci–dessus sont critiques pour la performance de l'algorithme fondé sur le FER. L'optimisation de ces paramètres dépend des liens et des applications, mais ils sont généralement fixés au moment de la conception. Ce mécanisme alourdit la performance des applications de type "stream" du fait que l'ajustement de la durée de la fenêtre pour obtenir des réponses rapides des applications de téléchargement produit des statistiques FER non fiables. De même, plusieurs trames sont transmises à un débit non optimal.

d) Contrôle de débit fondé sur les retransmissions

Une amélioration de l'approche fondée sur le FER consiste à rétrograder immédiatement lorsque la couche MAC lutte pour transmettre une trame correctement sur le lien. Ceci conduit à sélectionner le débit immédiatement inférieur après un petit nombre de retransmissions sans succès (généralement entre 5 et 10 retransmissions) [KAM97][VEG02]. Cette approche est implantée dans la partie "hardware" puisque le contrôle précis du débit fixé entre les retransmissions (de la même trame) est nécessaire. L'avantage de l'approche fondée sur les retransmissions vient du fait qu'elle combine un temps de réponse très court (peu de trames) pour traiter la détérioration des conditions du lien (rétrograder) avec une sensibilité faible aux débits des trafics.

L'algorithme de contrôle est en revanche plutôt pessimiste. Des pics d'erreur relativement courts causent de longues pertes dans le débit du fait que le passage aux débits supérieurs prend plus de temps que la rétrogradation étant donné qu'il est nécessaire de collectionner un FER significatif et prévenir les oscillations. Un autre inconvénient majeur de cette approche fondée sur les retransmissions est qu'en cas de collisions (lorsque les autres stations essaient de transmettre de façon simultanée), l'algorithme chute du fait de l'augmentation des retransmissions par trame. Ceci va d'abord causer une perte indésirable dans le débit (du fait du passage au débit inférieur), ce qui en fait va s'ajouter à la perte du débit causée déjà par la contention pour le médium. Deuxièmement, les chutes non nécessaires vers les débits faibles provoquent des injustices envers les autres utilisateurs car le temps d'émission additionnel réduit leur débit. Malheureusement sans l'utilisation du retour d'information CSI, il n'y a pas d'autre moyen pour l'algorithme de contrôle de distinguer les différentes causes de retransmissions des trames (les collisions ou le mauvais état du canal). Ainsi, cet algorithme ne peut pas éviter la rétrogradation non nécessaire vers des débits faibles en cas de collision sur le médium.

e) Contrôle de débit automatique fondé sur le SNR

Une limite fondamentale des retours d'informations indirectes fondés sur les statistiques est qu'ils classifient les conditions du lien comme bonnes ou mauvaises. Cette information binaire fournit des notions à propos de la direction à choisir pour fixer le débit de transmission, mais ne suffit pas pour sélectionner le débit optimal. Ceci conduit à un ajustement étape par étape lent dans le cas de changements brutaux de l'état du canal, et introduit le risque d'oscillation dans les conditions de stabilité. Une meilleure approche consiste à utiliser les mesures directes des conditions du lien comme par exemple le SNR (Signal to Noise Ratio). Le SNR est directement lié au taux d'erreur binaire du lien et de ce fait, au FER. En conséquence, le SNR est lié au délai paquet, à la gigue et au débit et détient le potentiel de fournir un retour d'information riche pour le contrôle automatique de débit[BAL99]. Connaissant le SNR courant et les courbes du débit par rapport au SNR pour chaque débit, le problème de sélection du débit est résolu instantanément : il suffit de commuter vers la valeur donnant le plus grand débit pour le SNR courant. Ceci peut être implanté de façon efficace au moyen de la table d'observation. Malgré cet avantage, le contrôle de débit fondé sur le SNR n'a pas été mis en pratique. Ceci provient principalement de trois raisons. Premièrement, pour certaines conditions du lien la relation entre le débit optimal et le SNR est grandement variable. Ceci est dû à l'imperfection des models qui décrivent le canal radio et aussi parce que la qualité du lien dépend aussi de bien d'autres paramètres. Deuxièmement, il n'est pas trivial d'obtenir une estimation fiable du SNR d'un lien. Plusieurs interférences radio fournissent une indication non calibrée de la puissance du signal (SSI = Signal Strength Indication). Troisièmement, le contrôleur de débit qui est du côté du processus d'envoi nécessite de connaître le SNR observé du côté de la réception.

Le problème de la communication de l'information retour sur le SNR est évoqué dans le standard émergeant 802.11h [IEE03], mais le standard n'a pas encore été finalisé. Le fonctionnement du standard 802.11h étant souhaité dans la bande des 5 GHz, il ne lui est pas possible d'être le successeur des standards tels que le 802.11b et le 802.11g qui opèrent à 2,4 Ghz. Ceci nécessite de définir de nouveaux suppléments. Un autre inconvénient du 802.11h est que l'information relatée du SNR est transmise en retour à l'aide d'une trame additionnelle de gestion. Cela va augmenter l'overhead MAC et peut causer le retard par non délivrance à temps de l'information sur le SNR du fait de la contention sur le médium. Plusieurs travaux utilisant l'information SNR pour le contrôle automatique de débit reposent sur des résultats de simulation et ne considèrent pas les difficultés liées à l'obtention d'une bonne estimation du SNR. Ces travaux sont concentrés sur l'option qui considère que le problème de bruit et du flottement du SNR peut être utilisé pour déterminer le débit adapté [BAL99][SAM98].

Une autre approche est discutée en [PAV03] dans laquelle une hypothèse est faite pour considérer le canal comme symétrique, ce qui signifie que le SNR observé à une station est très similaire à chaque point donné dans le temps. Cette hypothèse permet aux auteurs du document d'utiliser le SNR de la dernière trame ACK comme indicateur du SNR de l'autre côté, et l'utilisent pour sélectionner le débit auquel la prochaine trame de données peut être envoyé. Ils évitent le thème relatif à l'estimation du véritable SNR en dehors de la lecture du SSI en adaptant continuellement la table qui organise le SSI en débit pour les quatre débits de transmission de données du 802.11b. L'adaptation est nécessaire pour répondre à la variation des niveaux de bruit. L'analyse faite en [HAR05] révèle que, toutefois, l'algorithme de contrôle de débit de [PAV03] n'utilise pas le plein potentiel de l'information SNR, puisque l'adaptation prend place uniquement pour les retransmissions. En conséquence, l'adaptation ascendante n'est pas bien prise en charge. En outre, ils ne fournissent que des résultats de simulation et il n'est pas clair en pratique de voir comment leur algorithme se comportera. Finalement, ils ne discutent pas de l'aspect dynamique de leur approche, qui est, avec quelle rapidité le débit choisi atteint la valeur optimale pour certaines conditions données, lorsque ces conditions sont établies.

f) Contrôle automatique de débit hybride

Toutes les approches fondées sur les statistiques comme fondées sur le SNR ont chacune des avantages et des inconvénients. Celles reposant sur les statistiques offrent des performances robustes et maximisent le débit à long terme. Toutefois l'inconvénient majeur est sa réponse lente aux changements des conditions du lien, qui peut être une source de problème pour les applications temps réel. Le contrôle de débit fondé sur le SNR peut répondre très rapidement, mais du fait de l'incertitude et de la fluctuation de la relation entre l'information SNR et le BER (Bit Error Rate) du lien, il manque de stabilité et de fiabilité.

Une étape logique consiste alors à combiner les deux approches dans un algorithme hybride qui fournit à la fois de la robustesse et des réponses rapides. Le but poursuivi par les auteurs du document [HAR05] en combinant les méthodes basées sur les statistiques et celles reposant sur le SNR, est de supporter les applications de type "stream" en limitant les délais des paquets et la gigue autant que possible, même au dépend du débit si nécessaire. Cette option requiert une approche hybride puisque sous des conditions stables du lien, l'objectif est de fournir une communication robuste avec un débit élevé (approche fondée sur les statistiques), tandis que sous des conditions volatiles engendrées par exemple par le mouvement de l'utilisateur, l'objectif est d'éviter les retransmissions par des basculements rapides à des débits faibles (approche fondée sur le SNR). Un autre objectif tout aussi important décrit dans la formulation de l'algorithme hybride est de concevoir un système fonctionnel, d'où l'étude réalisée sur les problèmes pratiques associés aux méthodes reposant sur le SNR comme évoqué précédemment. Le contrôleur traditionnel reposant sur les statistiques est au cœur de l'algorithme hybride. Il s'agit d'un contrôleur de débit qui examine les débits adjacents pour déterminer si un changement de débit est nécessaire. La décision du contrôleur de cœur peut être supplantée par une seconde boucle de retour d'information. Cette boucle engage la portée acceptable de l'indication de la puissance du signal d'une trame acquittée (SSIA pour Signal Strength Indication of the Acknowledged frames) estimée pour chaque débit, sur la base de la courbe d'expression du débit par rapport au SNR. Ces bornes SSIA sont implantées comme une table de lecture indexée sur le débit.

I.4. Problématique de l'utilisation d'une méthodologie (Méthodes de conception)

La conception des protocoles réseaux reposant strictement sur le modèle en couches garantit un contrôle des interactions entre les couches. A partir de cette architecture, le développement et la maintenance d'une couche donnée se fait indépendamment des autres couches. Cette conception fournit un élément clé dans le succès d'Internet et sa prolifération. Pour conserver ces acquis et éviter le piège d'un mécanisme global complexe, la conception cross–layer doit passer par une méthodologie qui explicite toutes les interactions à mettre en œuvre ainsi que leurs implications.

Plusieurs raisons militent en faveur de la mise en place d'une méthodologie pour la mise en place de systèmes cross–layer. Il y a la nécessité de conserver les acquis du modèle en couches et ceux de la conception modulaire. Le contrôle des interactions entre les protocoles de différentes couches qui est l'un des avantages du modèle architectural ne peut être pleinement pris en compte dans la conception cross–layer que lorsqu'un outil standard de conception est mis en place, pour ressortir les implications de la mise en place d'une interaction donnée. Ce fait qui consiste à dessiner les contours des implications d'une interaction peut se faire à partir du simple formalisme ou graphe d'échange jusqu'à la description explicite de l'usage qui est fait de chaque interaction par chaque protocole impliqué. Si dans la description explicite de l'usage d'une interaction provenant de la méthode de conception, le principe de la compatibilité ascendante qui caractérise l'évolution des systèmes est pris en compte, l'implantation de chaque interaction au niveau d'un protocole donné ne doit pas être bloquante. Ceci implique que le protocole amélioré pourra aisément converser avec sa version précédente lorsque celle–ci ne supporte pas l'interaction implantée. Autrement dit, la mise en place d'une interaction peut être vue comme l'ajout d'une fonctionnalité supplémentaire, et de ce fait, la mise à jour du protocole ou de l'interaction se fait sans nécessité de "re–conception" globale, étant donné que l'objectif de la méthode est d'assurer la clarté de la conception. De même, en

explicitant le caractère d'indépendance de toutes les interactions, la méthode de conception permet également de tirer profit de l'effort parallèle en permettant aux concepteurs de se focaliser sur la mise en place d'une interaction donnée avec l'assurance que le système global fonctionnera normalement en intégrant les améliorations et les gains apportés par l'interaction. L'interaction peut elle même être considérée comme un module à part entière, quitte à appliquer le principe de subdivision récursif pour obtenir des sous–modules plus ou moins simplifiés.

Ainsi, pour conserver les acquis de l'architecture, de la conception modulaire et les apports de la conception cross–layer en terme de gain d'optimisation des performances, nous nous sommes penché vers un travail préalable dans lequel nous proposons une méthode efficace de conception de systèmes cross–layer. Cette méthode que nous appelons RCL (pour Reverse Cross–layer) fait l'objet du prochain chapitre.

I.5. Techniques disponibles

Nous avons procédé dans ce chapitre à l'étude de divers mécanismes cross–layer proposés. L'optimisation cross layer a été globalement prise en compte au niveau de toutes les couches. Les techniques relatives à l'utilisation de la notification explicite de congestion de TCP et de contrôle automatique de débit sont disponibles et sont à la mode dans le domaine d'échange d'information entre couches. Le travail complémentaire à réaliser pour compléter les diverses optimisations proposées se résume à leur intégration et à l'étude de leur comportement dans la pile complète de protocoles. La méthode de conception RCL que nous proposons au chapitre suivant permet également de répondre à cette préoccupation.

Chapitre II. La méthode de conception RCL (Reverse Cross–layer)

II.1. Introduction

Le passage de l'environnement filaire à l'environnement sans fil, très différents l'un de l'autre dans leurs caractéristiques propres, a conduit à considérer l'expansion des réseaux ad–hoc sous l'angle de techniques innovantes destinées à améliorer leurs performances. De cette orientation ont émergés les systèmes cross–layer avec pour objectif de fournir une adaptation efficace des protocoles du modèle en couches du réseau câblé à l'environnement sans fil. Divers travaux réalisés sur des protocoles pris isolement ont démontré les gains de performances pouvant être obtenus à travers le cross–layer. Il reste à assurer l'intégration globale sur une pile complète de protocoles et envisager la convertibilité des systèmes filaires vers les environnements sans fil. Cependant, étant donné la diversité des protocoles, leur différence de comportement (même s'ils appartiennent à la même couche) et les interactions qu'ils peuvent avoir entre eux, il convient de mettre en place une méthode de conception pratique qui s'adapte à toutes les interactions possibles assurant ainsi une évolution sans cesse croissante du modèle lorsqu'il intègre de nouveaux protocoles et de nouvelles interactions. Une interaction est un échange d'information entre des protocoles appartenant à des couches différentes, qui ne sont pas forcement contiguës, au niveau d'un même nœud du réseau, ou entre des nœuds distincts. L'architecture, que de telles interactions peuvent induire, peut être complexe, peut conduire à la conception d'un modèle partiel ou produire des modèles apparaissant comme antagonistes lorsqu'ils sont pris isolement. La méthode de conception permettra de produire dans un même modèle, différents aspects du modèle d'interactions cross–layer.

Dans le document [LI03] par exemple, les auteurs proposent le système MobileMan reposant sur le « full Cross–Layer design » qu'il oppose au « layer triggers » (signaux déclencheurs). Il ressort de l'application de la méthode RCL que nous proposons, que les signaux déclencheurs comme l'ECN (Explicit Congestion Notification) ou les L2 triggers sont une catégorie d'interactions cross–layer appartenant à la famille des actions atomiques « Cross–Layer » de "Notification". La méthode de conception RCL permettra d'établir qu'il s'agit de deux aspects différents du modèle global cross–layer, à savoir, une partie du modèle global consiste à la collection d'informations cross–layer qui seront mises à la disposition des couches, l'autre partie du modèle global consiste à échanger des messages ou des signaux entre les couches lorsque des événements particuliers surviennent par le simple fait que les instances des protocoles en liaison entre elles pour une communication réseau sont événementielles avec changement d'état.

Le document [RAM01] présente des mécanismes distincts d'implantation des interactions cross–layer comme par exemple l'usage de l'entête de paquet (méthode « Paket Header ») et la série de message du protocole ICMP (ICMP Messages). Ces mécanismes sont complémentaires lorsqu'ils sont pris individuellement pour des cas particuliers d'interaction.

La méthode de conception RCL a l'avantage de faire ressortir l'impact de chaque interaction cross–layer sur chaque protocole pour faciliter la mise à jour de son code source et l'adaptation de ce protocole au contexte cross–layer. Normalement, cette modification induit l'introduction d'un code de traitement de l'interaction qui permettra au protocole de fonctionner normalement même sans l'interaction (par le principe de la compatibilité ascendante).

La méthode peut être appliquée à une pile de protocoles particulière ou à un modèle conceptuel cross–layer existant dont l'évolution est envisagée pour intégrer une ou plusieurs nouvelle(s) interaction(s).

De façon générale, l'application d'une méthode de conception cross–layer doit permettre de produire ou de mettre à jour les modèles conceptuels cross–layer. Ces modèles conceptuels peuvent prendre divers formes. Ils permettent d'organiser efficacement la convertibilité des systèmes filaires dans l'environnement sans fil et font ressortir le travail de conversion à réaliser. Dans notre approche, ces modèles conceptuels sont : les modèles des interactions cross–layer et les tableaux descriptifs des interactions, tous étant produits par application de la méthode ascendante RCL.

Si nous considérons les modèles des interactions cross–layer comme étant du domaine conceptuel et les protocoles et les interactions elles–mêmes du domaine concret, nous désignons la méthode comme ascendante (d'où le qualificatif de "Reverse") en ce sens qu'elle permet d'évoluer du domaine concret vers le domaine conceptuel, pour une meilleure organisation et exploitation des potentialités permettant d'améliorer les performances des systèmes cross–layer ainsi conçus.

Nous présentons les sept étapes de cette méthode et les définitions des concepts qu'elle utilise au point II.2. Le point II.3 présente les résultats de l'application de la méthode au travers d'une pile de protocoles particulière, le recensement des actions atomiques utilisées dans la méthode, la production des tableaux d'interactions cross–layer, la production des modèles engendrés par ces interactions et la génération des tableaux descriptifs de ces interactions.

II.2. Méthode de conception RCL

II.2.1. Concept "d'Actions Atomiques" Cross–Layer (AACL)

Une Action Atomique Cross–Layer (AACL) peut être la mise à la disposition ou l'exploitation d'un paramètre d'une couche par d'autres couches. C'est aussi l'utilisation d'un service d'une couche qui intéresse d'autres couches ou tout simplement un comportement ou un événement survenu au niveau d'une couche qui doit être révélé à d'autres couches. Le terme atomique signifie que l'action ne doit pas être décomposable en une série d'autres actions, qui elles ont un impact sur une liste différente de protocoles. Par exemple, les actions comme « coordination du mécanisme de communication point à point de la couche liaison avec la communication de bout en bout de la couche transport » [WAN03] ou « exploitation de l'état du canal » ne sont pas atomiques, la première est imprécise, la deuxième fait référence à l'exploitation des paramètres d'environnement comme le BER, le SNR, la puissance de la porteuse ou la disponibilité du signal porteur. Elle peut également faire référence à la politique de retransmission, de gestion des acquittements, …

De façon générale, nous distinguons 3 types d'AACL dans notre modèle :

- Les AACL de "Mise à disposition" : ces AACL permettent d'exporter les valeurs des paramètres d'une couche, pour les rendre accessibles aux autres couches. Ces paramètres peuvent servir entre autres au contrôle d'admission de trafic ou à la QoS. Le système MobileMan, le système à base de serveurs WCI distribués sont conçus comme des modèles de mise à disposition.

- Les AACL de "Notification" : elles permettent de reporter un événement particulier survenant au niveau d'une couche, à destination d'une ou plusieurs autres couches. Comme exemple de telles interactions, nous noterons la coordination de contrôle d'erreur, la notification de la gigue dans la transmission d'un paquet du fait du mauvais état temporaire du canal (évitement d'envoi de nouvelles données à transmettre), la notification d'évitement de retransmission. Nous considérons le système CLASS comme un modèle d'actions de notification.

- Les AACL de services "Activables" : ce sont des modules complémentaires développés au niveau d'une couche et qui fournissent des paramètres ou des services pouvant intéresser d'autres couches.

II.2.2. Modélisation des interactions dans la méthode RCL

Les AACLs de "mise à disposition" et les AACLs de services "Activables" représentent des interactions locales à un nœud. Il s'agira des variables et des paramètres d'environnement portant des valeurs ayant des significations précises. Par exemple, lorsqu'un service est activé comme le VMAC [WAN03][VER01], une simple variable d'environnement booléenne signale cette activation. Des variables d'environnement complémentaires porteront des valeurs résultant de l'exécution de ce service qui les mettra à jour régulièrement (exemple : estimations locales des délais, des gigues, des collisions).

Les AACL de "Notification" comportent à la fois des interactions locales (par exemple la notification de la baisse significative du niveau d'énergie) [LI03] et des interactions distantes (par exemple la notification explicite de congestion) [RAM01].

C'est pourquoi notre modèle d'interactions Cross–Layer sera subdivisé en :

- un sous–système environnement : il comporte les variables et les paramètres d'environnement;

- un sous–système interface : il permet la communication entre des couches locales non contiguës;

- un sous–système distant : il permet la communication entre des couches appartenant à des nœuds différents.

Ces sous–systèmes fournissent une aide considérable au respect de l'architecture et à l'organisation du mode opératoire des interactions. Le sous–système environnement permet d'organiser les paramètres cross–layer utiles et simplifie leur mise à jour. Il facilite la mise en place des modules destinés à interpréter les valeurs de ces paramètres dont certains sont liés à un mécanisme à seuil. Ce principe centralisé allège la charge de la mise à jour à opérer au niveau des versions actuelles des protocoles. Les protocoles pourront se contenter de

notifications explicites provenant de ces modules. Les modifications du code source à apporter consistera à intégrer le traitement de ces notifications explicites, étant entendu que le protocole fonctionnera normalement même si le code de traitement de ces notifications n'est pas implanté. Le sous–système interface, quant à lui, organise les communications locales entre les différentes couches d'un nœud tandis que le sous–système distant organise les communications entre les couches appartenant à des nœuds différents. Les trois sous–systèmes ont des contenus évolutifs dépendant des interactions cross–layer mises en place.

Pour répondre à la nécessité de standardiser les mécanismes de communication du modèle global, nous proposons :

- pour le sous–système d'environnement : des fonctions d'entrée/sortie,

- pour le sous–système interface : il est possible d'effectuer le choix des fonctions d'entrée/sortie tout comme de choisir un protocole de communication à mettre en place, sachant que les informations véhiculées dans chaque interaction permettront de définir le protocole ou les fonctions d'Entrées/Sorties appropriées. En pratique, les deux solutions se rejoignent, étant donné que les protocoles sont implantés au moyen de fonctions d'Entrée/Sortie.

- pour le sous–système distant : les protocoles standards seront utilisés de façon spécifique pour chaque AACL.

II.2.3. Etapes de la méthode de conception RCL

La méthode RCL porte sur sept étapes conceptuelles qui permettent de créer des modèles conceptuels cross–layer. Pour rappel, les modèles conceptuels cross–layer permettent d'organiser efficacement la conversion des systèmes pour qu'ils intègrent les interactions efficaces en terme de performance entre différentes couches d'une pile protocolaire. Ils doivent permettre d'éviter le piège d'une conception complexe. La nécessité d'une clarté et d'une représentativité exhaustive de tous les éléments à prendre en compte dans la perception de la portée des modifications à apporter au système initial, impulsent à la méthode la dynamique de production de modèles conceptuels cross–layer qui prennent en compte ces contraintes. Les sept étapes de la méthodes sont décrites ci–dessous.

1. Choix de la pile de protocoles : il s'agit de fixer les protocoles du modèle en couches dont les modèles conceptuels des interactions cross–layer seront produits.

2. Recenser les AACLs : il peut s'agir d'un ensemble d'AACL ou d'une AACL unique pour laquelle il sera procédé à une évaluation de ses gains en performance.

3. Produire le tableau d'interaction des protocoles : le tableau présente en ligne les AACLs et en colonne les protocoles, à chaque case correspond l'un des symboles suivants : S (locale ou distante) pour signifier que le protocole est source de l'AACL, D pour signifier que le protocole est destinataire de l'AACL, U pour signifier que le protocole utilise la (les) donnée(s) de l'AACL, X pour signifier que les protocoles s'échangent des messages pour la mise en place ou l'exploitation de l'AACL. Le nombre qui suit le caractère "S" ou "D" indique la chronologie d'acheminement de l'information de l'AACL à travers la pile de protocoles.

4. Produire le tableau d'interaction des fonctions par protocole : les protocoles sont décomposés en fonction. Pour chaque protocole, le tableau précédent est repris, mais la colonne qui représente le protocole est fragmentée en autant de colonnes que de fonctions qui lui sont associées. Ce tableau a l'avantage de faire ressortir les fonctions à modifier au niveau de chaque protocole de la pile.

5. Déduire le modèle d'interaction par type d'AACL : à partir de la lecture du tableau produit à l'étape 3, il est possible de présenter un modèle d'interaction cross–layer pour chaque catégorie d'AACL. Ce modèle présente la pile des protocoles choisie à l'étape 1 à laquelle un sous–système complémentaire est associé avec des flèches qui matérialisent les interactions. Le modèle d'interaction a l'avantage d'aider à la compréhension des mécanismes internes cross–layer du modèle global.

6. Produire le tableau descriptif des interactions par protocole : dans ce tableau, les protocoles sont considérés un à un et le tableau d'interaction des fonctions par protocole est repris pour être traité ligne par ligne, conformément aux instructions qui suivent. Pour chaque AACL fixée et pour le protocole considéré, il sera mentionné pour davantage de lisibilité, la provenance de l'AACL, la fonction du protocole source ou destinatrice, le type de communication à utiliser (directe, via un sous–système, ascendante–descendante avec acheminement normal) et enfin l'exploitation qui peut être faite de l'AACL au niveau de la fonction du protocole concernée. C'est la spécification de cette exploitation qui détermine le travail d'implantation de l'AACL à réaliser.

7. Déduire les modalités d'implantation de chaque modèle d'interaction : chaque AACL appartient à un sous–système prédéfini ci–dessus. Chaque sous–système dispose de sa méthode de communication standardisée qui peut être des fonctions d'Entrées/Sorties ou des protocoles particuliers. Cette méthode de communication sera celle des modèles d'interaction à implanter qui ont été produits à l'étape 5.

NB : même si la lecture ou la présentation d'une même information se répète dans un format différent d'une étape à l'autre, la redondance donne davantage de clarté de conception à la méthode.

Avant d'expliciter en conclusion de ce chapitre les intuitions conceptuelles qui justifient chacune des étapes, nous allons d'abord appliquer la méthode RCL à un exemple de pile de protocoles, pour permettre de mieux appréhender les définitions des étapes présentées ci–dessus. L'avantage est que l'illustration de l'énonciation théorique à travers un exemple concret facilite la compréhension des étapes et des modèles qu'elles produisent.

II.3. Application de la méthode RCL

II.3.1. Choix de la pile des protocoles

Pour expérimenter notre méthode de conception de modèles cross–layer, nous avons choisi de l'appliquer à la pile de protocoles représentée par : le protocole TCP (Transmission Control Protocol) [RFC0793] au niveau transport, le protocole DSR (Dynamic Source Routing) [JET01] [JOH98] [DEM01] [DSR02] et le protocole IP (Internet Protocol) [RFC1349] [RFC0791] au niveau réseau, le protocole IEEE 802.11 [NI02] aux niveaux MAC et physique. Ces protocoles seront par la suite considérés au travers des tâches qu'ils assurent et qui permettent de déceler les fonctions qui sont les leurs dans le modèle en couches. L'objectif est d'aller à un niveau conceptuel plus fin, pour que parmi les fonctions d'un protocole donné, la méthode permette d'identifier celles qui sont modifiées par les actions atomiques cross–layer recensées.

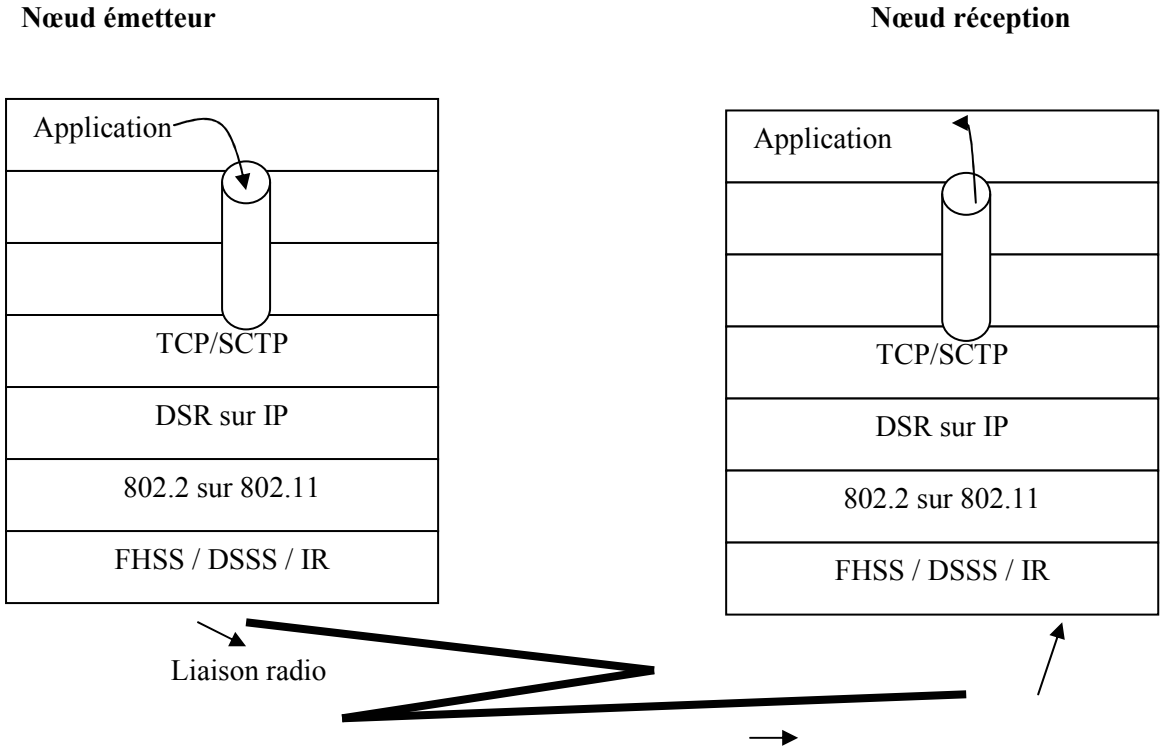


Figure II.1 : Exemple de pile de protocoles dans une transmission sans fil

II.3.2. Recensement des AACLs

A cette étape de recensement des AACL, nous considérons toutes les variables et les événements significatifs provenant de la pile de protocole choisie ainsi que les services qui peuvent être joints à chaque couche. Cette étape est celle du relevé exhaustif des interactions, étant entendu qu'en finalité, seules les interactions qui intéressent le cas de l'environnement réseau étudié seront retenues. En outre, les étapes ultérieures permettront de filtrer ces interactions pour qu'à l'implantation, ne soient retenues que les seules interactions qui engendrent un gain significatif. L'important à ce niveau est de trouver le ratio intuitif indicateur de l'intérêt de l'interaction. A titre de mesure permettant de jauger ce ratio, le concepteur doit mesurer le gain significatif apporté par rapport à la complexité des modifications ou de l'implantation de chaque interaction.

Dans les parties qui suivent, nous avons procédé au relevé exhaustif des AACL en considérant la subdivision préalable réalisée lors de la définition du concept des actions atomiques cross–layer. En rappel, nous avons défini trois catégories d'AACL à savoir les AACL de "Mise à disposition", les AACL de "Notification" et les AACL de services "Activables". Nous avons procédé au recensement des interactions potentielles tout en donnant un corps à chaque catégorie d'AACL.

II.3.2.1. AACL des services "Activables"

Le service DiffServ [WAN03] [RFC2998] est un service de la couche réseau que nous placerons dans la catégorie des services "Activable". Les réseaux DiffServ classifient les paquets en un petit nombre de flux agrégés ou "classes", fondés sur le code DiffServ (DSCP pour DiffServ CodePoint) dans l'entête IP du paquet. Ceci est connu comme une classification d'agrégat de comportement (BA : Behaviour Aggregate). Au niveau de chaque routeur DiffServ, les paquets sont soumis à un "comportement par saut" (PHB : Per–Hop Behavior) qui est invoqué par le DSCP. Le premier avantage de DiffServ est la facilité de passage à l'échelle. DiffServ élimine le besoin d'état par flux et de traitement par flux, et s'adapte bien aux grands réseaux.

Le service IntServ [WAN03] [RFC2998] est également un autre service de la couche réseau que nous considérons dans la catégorie des services "Activable". L'architecture des services intégrés (IntServ) définit un ensemble d'extensions du modèle traditionnel best effort de l'Internet avec l'objectif de fournir de la Qos aux applications. Un des composants–clés de l'architecture consiste en un ensemble de définitions de service, les services actuels sont le service à charge contrôlée et le service garanti. L'architecture suppose que des mécanismes explicites de signalisation soient utilisés pour acheminer l'information aux routeurs de telle sorte qu'ils puissent fournir les services demandés aux différents flux. Bien que RSVP (Resource ReSerVation Protocol) soit l'exemple le plus largement connu de mécanisme d'initialisation, l'architecture IntServ est conçue pour fonctionner avec d'autres mécanismes. Les services IntServ sont implantés au niveau des éléments du réseau tels que les nœuds individuels, les routeurs, les liens, les entités plus complexes comme le réseau ATM, 802.3, y compris les réseaux DiffServ.

Le service RSVP [RFC2998] est un autre service de la couche réseau que nous placerons dans la catégorie des services "Activable". RSVP est un protocole de signalisation que les applications peuvent utiliser pour demander des ressources du réseau. Le réseau répond par l'acceptation ou le rejet explicites des demandes. Dans le modèle courant d'utilisation de RSVP qui prévaut, RSVP signale des besoins en ressource par flux à l'élément réseau, en

utilisant les paramètres de IntServ. Ces éléments réseaux appliquent le contrôle d'admission de IntServ pour signaler les demandes. Des mécanismes de contrôle de trafic sur l'élément réseau sont configurés pour assurer que chaque flux accepté reçoive le service demandé en isolation stricte par rapport aux autres trafics. A cet effet, la signalisation RSVP configure les classificateurs de paquets de micro flux (MF) dans les routeurs qui prennent en charge le service IntServ tout au long du chemin du flux du trafic. Les mécanismes RSVP de politique de contrôle, de contrôle d'accès, d'authentification et de comptabilité ne sont disponibles que depuis peu.
RSVP est un exemple de service utilisable par les applications multimédia qui sont généralement à forte contrainte de délai dans la transmission des séquences multimédia associées à des intervalles de validité temporelle.

Nous considérons le FEC (Forward Error Correction) [WAN03] [RFC3453] introduit au niveau de la couche liaison comme un service "Activable" destiné à contrôler le taux d'erreur bit (BER ou Bit Error Rate). Ce service fait référence à la capacité de surmonter les pertes et la corruption des bits de données. En entrée du codeur FEC source, il y a le nombre k représentant le nombre de symboles sources (taille uniforme). Le codeur FEC génère un nombre de symboles codés qui sont de même taille que les symboles sources. Ces symboles codés sont placés dans des trames pour être transmis. Des informations suffisantes sont placées dans chaque paquet pour identifier les symboles particuliers codés transportés. Sur réception d'une trame contenant les symboles codés, le récepteur transmet ces symboles au décodeur FEC correspondant pour recréer une copie exacte des k symboles sources. Idéalement, le décodeur FEC peut recréer une copie exacte à partir de n'importe quel nombre k de symboles codés. Le code FEC est parfois utilisé en association avec un mécanisme ARQ pour les besoins de fiabilité.

Le mécanisme ARQ (Automatic Repeat Request) [WAN03] [RFC3453] [RFC3366] [RFC3155] introduit au niveau de la couche liaison sera également considéré dans la catégorie des services "Activables". Il est destiné à contrôler le taux d'erreur bit (BER). Le mécanisme ARQ fonctionne sur des blocs de données appelés trames (de petite taille fixée) pouvant contenir tout ou partie d'un paquet IP. Il opère des tentatives de délivrance de ces trames de la couche liaison de l'émetteur à la couche liaison du récepteur à travers le canal. Le mécanisme ARQ utilise une vérification d'intégrité pour chaque trame (exemple du strong link layer CRC) pour détecter les erreurs du canal et utilise un processus de retransmission par canal inverse pour renvoyer les trames perdues.
Pour permettre une utilisation efficace du canal, la taille maximale de la trame liaison doit être considérablement plus faible que la taille maximale des datagrammes IP spécifiée par la MTU d'IP. Une autre façon de réduire le taux de perte de trame est d'utiliser un codage comme par exemple le FEC. ARQ sur un simple lien dispose d'une boucle de contrôle plus rapide que la boucle de contrôle d'acquittement de TCP. Il peut fournir une retransmission plus rapide de segments TCP perdus sur le lien au moins pour un certain nombre de cas [RFC3155]. La procédure ARQ peut être capable d'utiliser des connaissances locales telles que le taux de transmission courant disponible, l'environnement d'erreur qui prévaut, la puissance de transmission disponible, pour optimiser la performance des transmissions relativement à l'état courant du lien.

Le service VMAC ou MAC virtuel [NI02] [VER01] sera considéré comme un service "Activable" de la couche liaison. C'est un système complémentaire qui permet d'observer le canal radio et d'établir des estimations locales des délais, des gigues, des collisions et des pertes de paquets au moyen de la mesure du temps libre DIFS, des paquets virtuels, des transmissions

simulées, de l'estampille des paquets virtuels. En utilisant les estimations de VMAC, une source virtuelle VS ajuste les paramètres de l'application et détermine le niveau de service pouvant être admis. Les estimations de VMAC et VS permettent de répondre aux applications temps réel ayant une exigence de délais qui doivent être courts et de priorité absolue.

II.3.2.2. AACL de "Mise à disposition"

Un principe simple permet au module système de gestion d'énergie [LI03] [GAL98] de mettre à jour le niveau d'énergie pour sa mise à disposition. Ce niveau d'énergie est par la suite exploité par les couches réseaux. Nous plaçons ce mécanisme dans la catégorie des AACL de "Mise à disposition". Les nœuds mobiles utilisent les sources d'énergie portables telles que les batteries faisant ainsi de la gestion d'énergie une contrainte pour la conception des réseaux ad–hoc. L'AACL courante matérialise l'interaction de mise à jour de la variable du sous–système d'environnement indiquant la jauge de la batterie pour que les protocoles adaptent leur comportement en fonction des différents seuils de cette variable. Pour illustrer le caractère fini de la source d'énergie, les travaux pionniers de Gallager [GAL98] définissent une communication fiable à travers une contrainte d'énergie en terme de capacité par unité d'énergie. Ces travaux indiquent qu'un réseau ad–hoc avec des nœuds disposant d'une énergie finie dispose seulement d'un nombre fini de bits qu'un nœud donné peut transmettre avant d'épuiser son énergie. L'allocation de ces bits aux différents besoins du réseau tels que la transmission d'information, l'échange d'informations de routage, l'acheminement des bits des autres nœuds, l'estimation du canal, …, devient un problème intéressant d'optimisation qui nécessite une coopération entre tous les processus. L'objectif de chaque protocole doit être d'utiliser le moins d'énergie possible.
A titre d'exemple illustratif, la couche MAC peut utiliser une stratégie de contrôle d'énergie en 3 modes : transmission à très faible énergie lorsque le canal est pauvre et le délai tolérable grand, transmission à forte énergie lorsque le canal et le délai sont moyens, transmission à très forte énergie lorsque la contrainte de délai est forte.

Il est possible de classer l'exposition du taux de perte de paquet [WAN03] par la fonction "contrôle" de la couche liaison parmi les AACL de "Mise à disposition". Cette AACL matérialise le calcul fait au niveau de la couche liaison et la mise à jour des variables du sous–système environnement dont les formules de calcul établies par unité de temps sont les suivantes :
Nombre de paquets non acquittés / nombre total de paquets envoyés dans l'unité de temps
Nombre de paquets reçus endommagés / nombre total de paquets reçus dans l'unité de temps
Le premier paramètre donne le taux de perte de paquet à l'émission, le second le taux de perte de paquet à la réception.
Les valeurs par seuil de chacun des paramètres permettent de définir l'état "bon" ou "mauvais" du canal.

Nous considérons également l'AACL de "Mise à disposition" du SNR (Signal to Noise Ratio) par la couche physique [WAN03] pour matérialiser la mise à jour de la variable correspondante du sous–système environnement par la couche physique. Cette variable est un paramètre qui donne la valeur du rapport signal à bruit valable pendant un certain temps ou jusqu'au prochain changement de valeur. La valeur par seuil de ce paramètre permet de définir l'état "bon" ou "mauvais" du canal.
Pour permettre d'appréhender l'expression du SNR, il est important de considérer les définitions des termes de la fraction qui sont utilisés dans le calcul de sa valeur, le numérateur et le dénominateur étant le signal et le bruit. Un signal est une énergie détectable transmise et pouvant être utilisée pour transporter une information. Un bruit est un dérangement à travers la bande de fréquence. Le SNR exprimé généralement en décibels est le ratio de l'amplitude d'un signal donné par rapport à l'amplitude du bruit à un instant donné.

De la même façon, nous considérons l'AACL de "Mise à disposition" du RSS (Received Signal Strength) par la couche physique [WAN03][MAZ88] pour matérialiser la mise à jour de la variable correspondante dans le sous–système environnement par la couche physique. Cette variable est un paramètre qui donne l'intensité du signal reçu à partir d'un nœud du réseau et est valable un certain temps. La valeur par seuil de ce paramètre permet d'évaluer approximativement la distance qui sépare le nœud émetteur du nœud destinataire, ou d'établir l'accessibilité à portée directe, pour les besoins des protocoles de routage.
Pour illustrer l'importance du paramètre RSS dans le contexte d'émissions simultanées sur un lien sans fil, dans le document [MAZ88], l'auteur fait référence à un modèle du phénomène de capture fondé sur une discrétisation de la puissance captée par la station de base en p niveaux indicés de 1 à p par puissance reçue décroissante. Il y est supposé que la capture a toujours lieu pour des émissions simultanées appartenant à des niveaux de puissance différents et n'a jamais lieu à l'intérieur d'un même niveau pour cause de collision. Dans la réalité, la capture a lieu lorsque le rapport signal/bruit de l'émission la plus puissante sur la somme de toutes les autres est supérieur à un certain seuil qui n'est fonction que de la sophistication du récepteur de la station de base.

La "Mise à disposition" du taux d'erreur bit BER (Bit Rate Error) par la couche physique [WAN03] sera considérée comme faisant partie des AACL de cette catégorie. Cette AACL matérialise le calcul fait au niveau de la couche physique et la mise à jour de la variable du sous–système environnement dont la valeur est donnée par unité de temps par la formule suivante :
Nombre de bits reçus endommagés / nombre total de bits reçus dans l'unité de temps
Les valeurs par seuil de ce paramètre permettent de définir l'état "bon" ou "mauvais" du canal. Ce paramètre est intimement lié au calcul du taux de perte de paquets de la couche liaison.

II.3.2.3. AACL de "Notification"

La gigue d'envoi des paquets établie au niveau de la couche liaison [WAN03] du fait des retransmissions ou du fait de la persistance des paramètres indiquant le mauvais état du canal (taux de perte de paquet, SNR, BER) est utilisable par TCP au niveau transport et par la couche application. Nous placerons l'interaction consistant à notifier ce décalage de transmission dans la catégorie des AACL de "Notification".

Pour des raisons de saturation, de handoff de la couche IP [WAN03] [MIN02] ou divers autres raisons nécessitant le gel des retransmissions, de l'admission d'un nouveau trafic, une interaction directe de la couche liaison peut être adressée à TCP (couche transport) et à la couche application. Nous considérons l'évitement de retransmission dans la catégorie des AACL de "Notification".

Lors d'une transmission d'un nœud émetteur vers un nœud destinataire à travers la pile des protocoles TCP, DSR, IP, 802.11, un segment TCP est encapsulé dans un paquet DSR qui lui même est encapsulé dans un datagramme IP. La couche liaison utilise une ou plusieurs trames pour transmettre sur le lien physique un datagramme IP. Dans le schéma du 802.11, la couche physique implante des acquittements par exploitation des intervalles SIFS [NI02]. Le DSR est conçu pour utiliser ces acquittements sur la base du regroupement au niveau de la couche liaison des trames acquittées contenant le datagramme IP intégral [JET01][JOH98]. Nous étendrons l'exploitation de ces acquittements jusqu'à la couche transport avec une AACL de "Notification" d'acquittement.

Les interactions ECN [SHA03] [RAM01] et ELN [WAN03] [BAL98] sont des AACL de "Notification". Lorsque les routeurs détectent une congestion avant le débordement des files internes, ils positionnent le bit ECN de l'entête du segment TCP. Le destinataire reporte cette indication de congestion à l'expéditeur, qui en recevant un tel paquet avec le bit ECN positionné, invoque son mécanisme d'évitement de congestion. Dans le cas des réseaux sans fil avec infrastructure, un module appelé agent snoop peut être introduit au niveau de la station de base pour enregistrer tous les segments des connexions TCP et conserver les traces de trous (segments TCP non acquittés par le récepteur et qui sont perdus sur le lien sans fil). L'agent snoop positionne le bit ELN dans un ACK dupliqué lorsqu'il correspond à un segment du trou avant de l'acheminer à l'expéditeur. En recevant un tel ACK, l'expéditeur retransmet le segment et ne prend aucune action de contrôle de congestion.
Etant donné que notre modèle cible porte dans l'environnement des réseaux ad–hoc et que l'utilisation de l'agent snoop n'est pas appropriée sur un nœud mobile du fait de l'impossibilité de discerner les pertes dues à la corruption et celles dues à la congestion dans le réseau ad–hoc, l'AACL qui se rapporte à l'usage de l'ELN ne sera pas utilisée.

L'AACL de "Notification" de la baisse significative du niveau d'énergie [LI03] tire sa source du module système de gestion d'énergie. Lorsque le niveau d'énergie disponible atteint un seuil critique, le module système de gestion d'énergie envoie l'information à toutes les couches. Cette interaction conduit à diverses conséquences. La couche MAC peut éliminer autant que possible les collisions du fait que les retransmissions induisent plus de consommation d'énergie, elle peut passer en mode "standby" (mode veille) pour économiser l'énergie. La couche réseau par exemple peut utiliser des métriques pour un routage sensible à l'énergie : minimiser l'énergie consommée par paquet, maximiser le temps de partition du réseau, minimiser la variance en niveau de puissance du nœud, minimiser le coût par paquet, minimiser le coût maximum de nœuds. La transmission de paquet avec des entêtes plus compactes est une autre technique efficace de gestion d'énergie. Au niveau de la couche transport, la répétition des retransmissions occasionne plus de consommation d'énergie. Elle doit être évitée autant que possible tout en maintenant un certain niveau de performance de la communication.

L'AACL de "Notification" pour la récupération d'un paquet [JET01][JOH98] repose sur l'exploitation des fonctionnalités du protocole DSR. En effet, le protocole DSR est conçu pour détecter au niveau d'un nœud intermédiaire et à travers la maintenance de route (fondée sur les acquittements) qu'un lien de la route source du paquet est hors d'usage. C'est pourquoi, lorsque le nœud intermédiaire dispose d'une autre route pour l'adresse IP de destination du paquet en cours d'acheminement, il modifie la route source du paquet et incrémente la valeur du champ "Récupération" avant l'envoi du paquet au prochain saut de la nouvelle route. Le nœud renvoie également une erreur de route à l'émetteur original du paquet.
L'opportunité d'une telle récupération se traduit également lorsque la couche liaison établit l'inaccessibilité du prochain nœud dans la transmission relativement à des paramètres d'état du canal comme le BER, le RSS.

Dans la catégorie des AACL de "Notification", nous considérons le principe de notification de la puissance du signal reçu à partir d'un nœud [WAN03][MAZ88] disponible au niveau de la couche liaison du 802.11. Cette AACL sur un nœud courant matérialise l'envoi au DSR par la couche liaison 802.11 de la valeur du RSS d'un paquet. Cette valeur doit être prise en compte dans la table de routage en engendrant au besoin sa mise à jour selon la mobilité du nœud émetteur et du nœud courant. L'interprétation de la valeur du RSS permet d'établir l'accessibilité à portée directe ou pas d'un nœud en vis à vis.

L'AACL de "Notification" de la gigue d'envoi due à la défaillance d'une route [JET01][JOH98] repose sur l'exploitation des messages de la maintenance de route du protocole DSR. Cette AACL permet de poursuivre l'adaptation du protocole TCP au contexte des réseaux ad–hoc dont la mobilité des nœuds occasionne des pertes de paquets dues au changement fréquent de route. L'AACL matérialise l'interaction à travers laquelle le DSR informe TCP d'une erreur de route survenue dans l'acheminement d'un paquet pour éviter que TCP n'invoque son mécanisme de contrôle de congestion à l'expiration de son temporisateur de retransmission. Dans la fonction de routage qu'il assure, le DSR est conçu pour recevoir et traiter des messages d'erreur de route.

L'AACL de "Notification" de la gigue d'envoi due au changement de route [JET01][JOH98] est une autre possibilité offerte par l'exploitation des messages de la maintenance de route du protocole DSR. Cette AACL permet également de poursuivre l'adaptation du protocole TCP au contexte des réseaux ad–hoc caractérisés par la mobilité des nœuds et la perte de paquets du fait du changement fréquent de route. Cette interaction de DSR avec TCP permet d'informer ce dernier d'une modification de la route source survenue dans l'acheminement d'un paquet pouvant prolonger les délais de réception, pour éviter l'invocation par TCP du mécanisme de contrôle de congestion. DSR est conçu pour recevoir et traiter des messages d'erreur de route suite à une récupération d'un paquet par un nœud intermédiaire de la route source.

II.3.3. Tableau des interactions des protocoles par AACL

Après la fin de l'étape de recensement des AACL et de leur classification par catégorie, la méthode RCL permet de produire le tableau des interactions des protocoles par AACL à l'étape courante.

À cette étape 3 de l'application de la méthode RCL, les AACL recensées ci–dessus seront organisées dans le tableau des interactions des protocoles. Ce tableau comporte les protocoles qui utilisent ces AACL, la source et la destination de chaque interaction.
Pour faciliter la lecture de ce tableau, il faut noter par exemple que la gigue d'envoi de paquet est établie par la couche liaison (du fait des retransmissions ou de la persistance des paramètres indiquant le mauvais état du canal tels que le taux de perte de paquet, le SNR, le BER) et est envoyée au protocole TCP. Dans un autre exemple, le protocole IP d'un nœud distant est la source de l'AACL de notification explicite de congestion qui est destinée au protocole TCP.

A travers l'AACL de mise à disposition du SNR par la couche physique, le protocole IEEE 802.11 met à jour le paramètre du sous–système environnement qui indique la valeur du ratio signal à bruit. Ce paramètre est utilisé par la couche liaison, par le protocole TCP et par la couche application.

La première source de l'AACL de notification de la puissance du signal reçu est la couche physique 802.11 qui informe la couche liaison, qui à son tour envoie cette information au protocole DSR. Un dernier exemple de lecture du tableau des interactions des protocoles se rapporte à l'activation du service ARQ par la couche liaison 802.11, service qui est par la suite utilisé par le protocole TCP relativement au tableau de description des interactions de la sixième étape de la méthode RCL.

Action Atomique Cross–Layer (AACL)	Protocoles					
	Application	**TCP**	**DSR**	**IP**	**Liaison 802.11**	**Physique 802.11**
"Notification" de la gigue d'envoi des paquets		D			S	
"Notification" d'évitement de retransmission		D	D		S	
"Notification" d'acquittement		D3	D2, S3		D1 ,S2	S1
"Notification" explicite de congestion		D		S distante		
"Notification" de la baisse significative du niveau d'énergie	D	D	D	D	D	S
"Notification" pour la récupération d'un paquet			D		S	
"Notification" de la puissance du signal reçu à partir d'un nœud			D2		D1, S2	S1
"Notification" de la gigue d'envoi due à la défaillance d'une route		D	S			
"Notification" de la gigue d'envoi due au changement de route		D	S			
"Mise à disposition" du taux de perte de paquet	U	U			S	
"Mise à disposition" du SNR (Signal to Noise Ratio)	U	U			U	S
"Mise à disposition" du RSS (Received Signal Strength)	U	U			U	S
"Mise à disposition" du taux d'erreur bit BER (Bit Rate Error)	U	U			U	S
"Mise à disposition" du niveau d'énergie	U	U	U	U	U	S
Service RSVP "Activable" de contrainte de délai	X			X	X	
Service VMAC "Activable"	U/X				U/X	S
Service IntServ "Activable"	U/X			S/X		
Service DiffServ "Activable"	U/X			S/X		
Service FEC (Forward Error Correction) "Activable"		U			S	
Service ARQ (Automatic repeat request) "Activable"		U			S	

Table II.1. Tableau des interactions des protocoles par AACL.

Légende :
X : échange bidirectionnel S : Source de l'AACL D : Destination de l'AACL U : utilise les données de l'AACL
nombre : indique la chronologie d'acheminement de l'interaction à travers les couches.

II.3.4. Tableau d'interaction des fonctions par AACL

II.3.4.1. Cas du protocole TCP

Le protocole TCP assure un transfert fiable de données à travers une connexion entre deux stations. Il assure une fonction de contrôle de données transférées qui permet d'établir si les données transférées sont endommagées, perdues ou dupliquées ou encore dé-séquencées en cas de livraison dans un mauvais ordre. Le protocole assure également d'autres fonctions dont la fonction de correction d'erreur à travers le mécanisme de retransmission, la fonction de contrôle de flux grâce au mécanisme d'usage de la fenêtre de transmission, la fonction de contrôle de congestion grâce au mécanisme d'évitement de congestion et aussi la fonction de gestion de priorité par le mécanisme de transmission prioritaire des données.

Le tableau suivant représente le modèle conceptuel d'interaction entre les fonctions de TCP et les AACL. Il présente les AACL utilisées par les différentes fonctions du protocole TCP ainsi que les autres protocoles qui interviennent. L'exemple de l'AACL de notification de la gigue d'envoi des paquets indique que la fonction TCP de contrôle de données transférées est la destination effective de cette AACL initiée par le protocole 802.11 à la couche liaison. La modification envisagée pour que cette fonction prenne en compte le traitement de l'AACL est décrite au tableau de description des interactions de l'étape six de la méthode.

Un autre exemple présente la source IP distante comme étant la source de l'AACL de notification explicite de congestion. La fonction TCP de contrôle de congestion en est la fonction destinatrice.

La notification d'acquittement est une AACL qui prend sa source à partir du mécanisme de transmission de la couche physique 802.11 et est établie à l'échelle d'une trame par la couche liaison pour ensuite être acheminée jusqu'au protocole DSR et enfin jusqu'à la fonction de contrôle de données transférées de TCP.

Actions Atomiques Cross–Layer	Fonctions TCP					Autres Protocoles			
	Contrôle données transféré.	**Correct° d'erreur**	**Contrôle de congest°**	**Gestion des priorités**	**Gest° de flux**	**DSR**	**IP**	**Liaison 802.11**	**Physique 802.11**
"Notification" de la gigue d'envoi des paquets	D							S	
"Notification" d'évitement de retransmission	D					D		S	
"Notification" d'acquittement	D3					D2, S3		D1, S2	S1
"Notification" explicite de congestion			D				S distante		
"Notification" de la baisse significative du niveau d'énergie	D					D	D	D	S
"Notification" de la gigue d'envoi due à la défaillance d'une route	D					S			
"Notification" de la gigue d'envoi due au changement de route	D					S			
"Mise à disposition" du taux de perte de paquet	U							S	
"Mise à disposition" du SNR	U							U	S
"Mise à disposition" du RSS	U							U	S
"Mise à disposition" du taux d'erreur bit BER	U							U	S
"Mise à disposition" du niveau d'énergie	U					U	U	U	S
Service FEC "Activable"	U							S	
Service ARQ "Activable"		U						S	

Table II.2. Tableau d'interaction des fonctions TCP par AACL

II.3.4.2. Cas du protocole DSR

Le protocole DSR assure une fonction de routage par utilisation de messages de route, des données de la couche IP et des structures de routage. Il assure également diverses autres fonctions dont :

- une fonction de découverte de route au moyen de messages de requête de route et de réponse de route,
- une fonction de contrôle de transmission faisant partie de la maintenance de route par utilisation des acquittements,
- une fonction de gestion d'erreur de route liée à la maintenance de route au moyen de message d'erreur de route,
- une fonction de récupération de paquets au travers de la maintenance de route par la modification de la route source d'un paquet et la notification à l'émetteur,
- ainsi qu'une fonction de segmentation pour ajuster la taille des paquets à celle des chemins.

Le tableau ci–dessous représente la modélisation conceptuelle des interactions entre les fonctions du protocoles DSR et les AACL recensées à l'étape 2 de la méthode RCL. Ce tableau présente les AACL utilisées par le protocole DSR, les fonctions du protocole DSR et les autres protocoles qui interagissent.

Pour expliciter l'utilisation des AACL par chaque fonction de DSR, prenons l'exemple qui indique que la fonction de contrôle de transmission du DSR est la destination effective de l'AACL de notification d'évitement de retransmission envoyée par le 802.11 de la couche liaison. Cette fonction sera modifiée pour prendre en compte le traitement de l'AACL selon le tableau de description des interactions de l'étape six de la méthode.

La fonction de routage est la destination effective de l'AACL de notification de la puissance reçue à partir d'un nœud. La valeur de cette puissance est établie d'abord à la couche physique 802.11 pour être acheminée à la couche liaison et enfin au protocole DSR pour être utilisée par sa fonction de routage.

Actions Atomiques Cross–Layer	Fonctions DSR						Autres Protocoles			
	Routage	**Découv. de route**	**Ctrl de trans.**	**G. erreur de route**	**Récupé -ration**	**Segmen -tation**	**Applicat-ion**	**TCP**	**Liaison 802.11**	**Physique 802.11**
"Notification" d'évitement de retransmission			D					D	S	
"Notification" d'acquittement			D2, S3					D3	D1, S2	S1
"Notification" de la baisse significative du niveau d'énergie		D	D		D	D	D	D	D	S
"Notification" pour la récupération d'un paquet					D				S	
"Notification" de la puissance du signal reçu à partir d'un nœud	D2								D1, S2	S1
"Notification" de la gigue d'envoi due à la défaillance d'une route				S				D		
"Notification" de la gigue d'envoi due au changement de route					S			D		
"Mise à disposition" du niveau d'énergie		U	U		U	U	U	U	U	S

Table II.3. Tableau d'interaction des fonctions DSR par AACL

II.3.4..3. Cas du protocole IP

Le protocole IP assure un transfert de données en mode datagramme à travers :

- la fonction de routage par la vérification de l'entête du datagramme IP, la vérification de la durée de vie du datagramme, l'identification des stations, également par l'utilisation des algorithmes et des structures de routage
- la fonction de segmentation pour ajuster la taille des paquets à celle des chemins.

Le tableau d'interaction des fonctions IP par AACL ci–dessous présente les AACL utilisées par les fonctions du protocole IP auxquelles sont associées les autres protocoles qui interviennent dans ces interactions. La fonction de routage à laquelle se greffe le module RSVP interagit avec la couche application pour que ce service soit activé par la couche réseau. Après son activation, ce service est utilisé par la couche application. L'usage fait par chaque fonction de l'AACL est décrit dans le tableau de description des interactions de l'étape six de la méthode RCL.

La fonction de routage d'une source IP distante est la fonction émettrice de l'AACL de notification explicite de congestion qui est par la suite acheminée à destination du protocole TCP.

Actions Atomiques Cross–layer	**Fonctions IP**		**Autres Protocoles**			
	Routage	**Segmentation**	**Application**	**TCP**	**Liaison 802.11**	**Physique 802.11**
"Notification" explicite de congestion	S distante			D		
"Notification" de la baisse significative du niveau d'énergie	D	D				
Service RSVP "Activable" de contrainte de délai	X		X		X	
Service IntServ "Activable"	X		X			
Service DiffServ "Activable"	X		X			
"Mise à disposition" du niveau d'énergie	U	U				

Table II.4. Tableau d'interaction des fonctions IP par AACL

II.3.4.4. Cas du protocole liaison802.11

Le protocole IEEE 802.11 assure une fonction de contrôle à la sous–couche LLC (protocole 802.2), une fonction de sécurité et d'intégrité au niveau de la sous–couche MAC (protocole 802.10), une fonction de contrôle d'accès au médium sans fil (802.11) à la sous–couche MAC.

Les AACL utilisées par les fonctions du protocole IEEE 802.11 ainsi que les autres protocoles qui interviennent dans ces interactions sont présentés dans le tableau d'interaction ci–dessous. A titre d'exemple de lecture de ce tableau, il ressort que la fonction de contrôle du 802.11 est la fonction source de l'AACL de notification de la gigue d'envoi des paquets envoyée au protocole TCP.

Actions Atomiques Cross–layer	Fonctions couche liaison			Autres Protocoles				
	Contrôle	Sécurité intégrité	Accès au médium	Applicat-ion	TCP	DSR	IP	Physique 802.11
"Notification" de la gigue d'envoi des paquets	S				D			
"Notification" d'évitement de retransmission	S				D	D		
"Notification" d'acquittement	D1, S2				D3	D2, S3		S1
"Notification" de la baisse significative du niveau d'énergie	D			D	D	D	D	S
"Notification" pour la récupération d'un paquet	S					D		
"Notification" de la puissance du signal reçu à partir d'un nœud	D1, S2					D2		S1
"Mise à disposition" du taux de perte de paquet	S			U	U			
"Mise à disposition" du SNR	U			U	U			S
"Mise à disposition" du RSS	U			U	U			S
"Mise à disposition" du taux d'erreur bit BER	U			U	U			S
"Mise à disposition" du niveau d'énergie	U			U	U	U	U	S
Service VMAC "Activable"	U/X			U/X				S
Service FEC "Activable"	S				U			
Service ARQ "Activable"	S				U			

Table II.5. Tableau d'interaction des fonctions 802.11 par AACL.

II.3.5. Déduction de(s) modèle(s) d'interaction des AACL

Les modèles d'interaction des AACL répartis selon chacune des 3 catégories d'AACL, peuvent être déduits à cette étape de l'application de la méthode RCL. Ces modèles d'interaction ont pour but d'expliciter l'utilisation qui est faite des sous–systèmes engendrés par les interactions ainsi que les mécanismes cross–layer internes aux couches réseaux. Ils permettent également d'appréhender la complexité des interactions à mettre en œuvre.

Pour faciliter la lisibilité, le sous–système interface et le sous–système distant ne seront pas représentés dans les modèles ci – dessous, mais serons implicites.

II.3.5.1. Cas des AACLs de "Notification"

Le modèle d'interaction des AACL de "Notification" est représenté à la figure II.1 ci dessous. Il comporte la pile des protocoles étudiée ainsi que le gestionnaire d'énergie du système et la matérialisation du protocole TCP distant. Dans cette représentation, les numéros représentent les AACL données en légende, les cercles décrivent les couches sources des AACL et les flèches matérialisent les couches destinatrices des AACL. Par exemple, le chiffre 5 indique que lorsque le niveau d'énergie atteint un seuil critique, le gestionnaire d'énergie du système diffuse cette information à destination de toutes les couches.

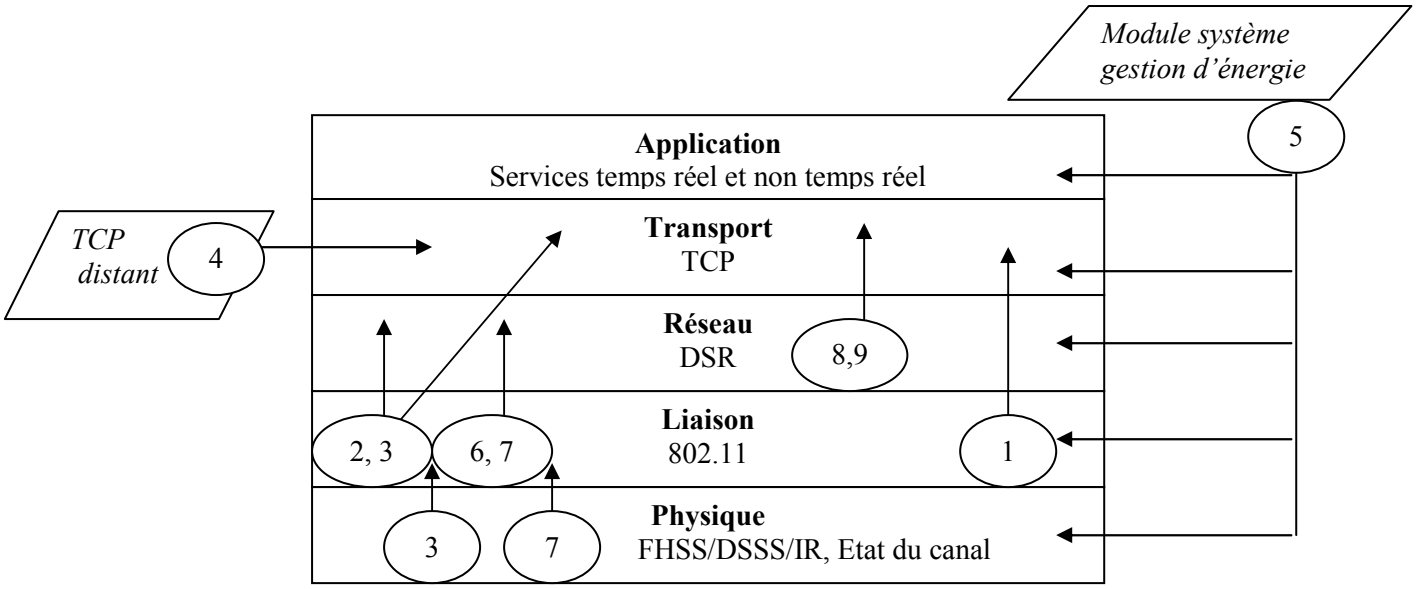


Figure II.2 : Modèle d'interaction des AACL de "Notification".

Légende :

1. "Notification" de la gigue d'envoi des paquets
2. "Notification" d'évitement de retransmission
3. "Notification" d'acquittement
4. "Notification" explicite de congestion
5. "Notification" de la baisse significative du niveau d'énergie
6. "Notification" pour la récupération d'un paquet
7. "Notification" de la puissance du signal reçu à partir d'un nœud
8. "Notification" de la gigue d'envoi due à la défaillance d'une route
9. "Notification" de la gigue d'envoi due au changement de route

II.3.5.2. Cas des AACL de "Mise à disposition"

Le modèle d'interaction ci–dessous est donné par les AACL de mise à disposition relativement à la pile de protocoles, tout en impliquant le sous–système environnement et le gestionnaire d'énergie du système. Dans ce modèle d'interaction et relativement à l'étape de recensement des AACL, le protocole DSR utilise uniquement la variable indiquant le niveau de la batterie régulièrement mise à jour par le gestionnaire d'énergie du système. Les numéros utilisés dans ce modèle représentent les AACL données en légende, chaque cercle contient la liste des AACL concernées individuellement par l'opération indiquée par la flèche. La flèche provenant du (respectivement allant vers le) sous–système environnement indique une opération de lecture (respectivement de mise à jour) des paramètres de ce sous–système. Par exemple, le numéro 1 indique que la couche liaison 802.11 met à jour le taux de perte de paquets qui est utilisé par le protocole TCP et la couche application indépendamment des autres AACL de la liste.

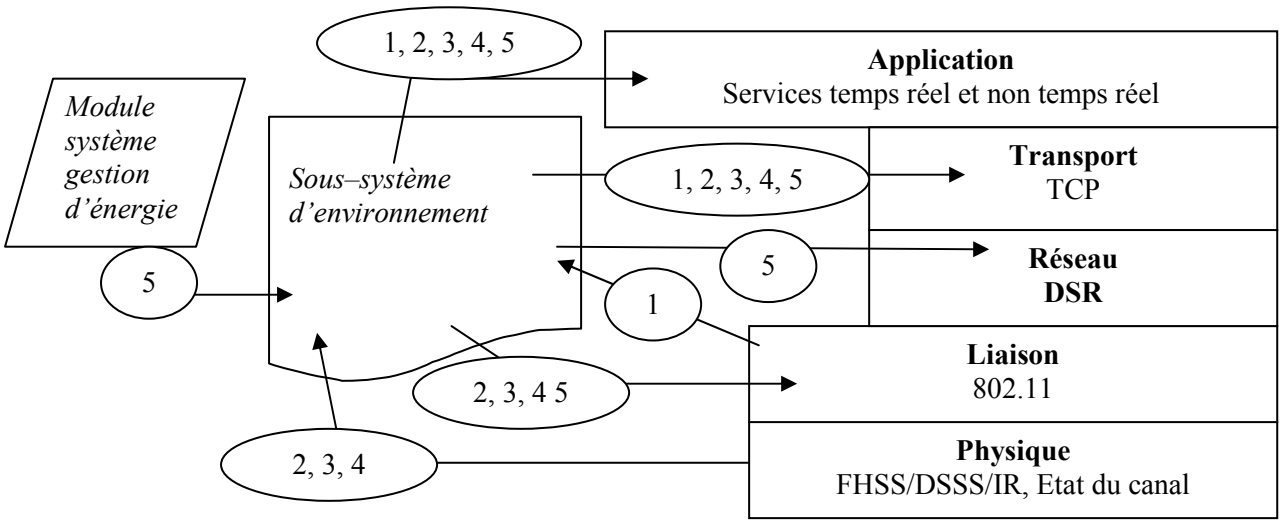


Figure II.3 : Modèle d'interaction des AACL de "Mise à disposition"

Légende :

1. "Mise à disposition"/utilisation du taux de perte de paquet
2. "Mise à disposition"/utilisation du SNR (Signal to Noise Ratio)
3. "Mise à disposition"/utilisation du RSS (Received Signal Strength)
4. "Mise à disposition"/utilisation du taux d'erreur bit BER (Bit Rate Error)
5. "Mise à disposition"/utilisation du niveau d'énergie

II.3.5.3. Cas des AACL "Activables"

Par le même mécanisme de déduction des modèles d'interaction que précédemment, le modèle donné par les AACL de services activables relativement à la pile de protocole et au sous–système environnement est présenté ci–dessous. A partir ou envers le sous–système environnement, les numéros, les flèches et les cercles ont la même signification que le modèle d'interaction des AACL de "Mise à disposition". La différence provient de trois cas dans lesquels les doubles flèches indiquent l'échange d'information entre les couches. Cet échange se fait à travers le sous–système interface qui est implicite dans ce modèle pour des raisons de lisibilité. Ces échanges permettent d'activer les services des couches appropriées.
Par exemple, les applications demandent l'activation des services aux couches concernées. Une fois les services activés, le sous–système environnement est renseigné. Les autres couches utilisent les indicateurs d'activation ou les paramètres des services activés pour adapter leur comportement.

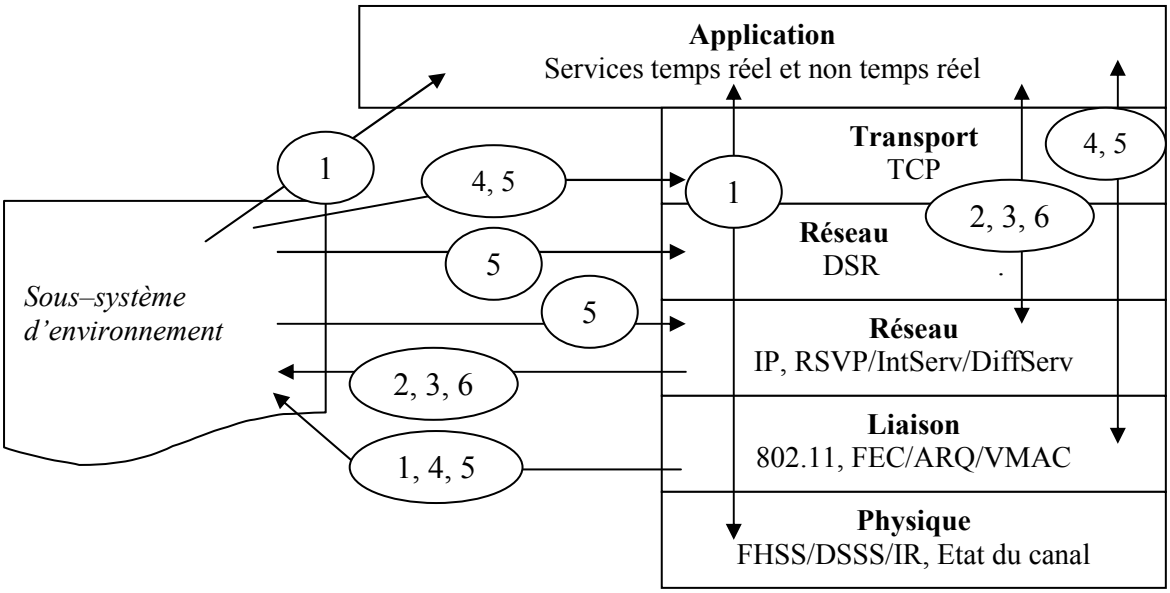


Figure II.4 : Modèle d'interaction des AACL "activables"

Légende : Positionner l'indicateur d'activation, lire les paramètres ou utiliser le service :

1. VMAC
2. IntServ
3. DiffServ
4. FEC
5. ARQ
6. RSVP de contrôle coordonné de délai

II.3.6. Tableau descriptif des interactions

A cette sixième étape de l'application de la méthode RCL, il nous est possible de déduire le tableau de description des interactions de chaque protocole de la pile considérée à l'étape 1. Chaque tableau permet d'expliciter l'exploitation qui peut être faite de chaque AACL par la fonction de chaque protocole concernée.

Pour faciliter la lisibilité du document, nous limiterons la présentation du tableau descriptif des interactions à trois colonnes à savoir l'AACL, la fonction du protocole et l'exploitation qui peut être faite de l'AACL par la fonction.

II.3.6.1. Cas du protocole TCP

Le tableau de description des interactions de TCP qui suit indique l'utilisation de chaque AACL faite par les fonctions du protocole TCP et les modifications du code source de ces fonctions que nous proposons.

AACL	Fonction TCP	Exploitation au niveau du TCP
"Notification" de la gigue d'envoi des paquets	Contrôle données transférées	S'il y a dépassement possible du temps d'attente d'ACK, réinitialiser le compteur d'attente d'ACK du segment. Pas de retransmission de ce segment pour la nouvelle durée. Ne pas invoquer le mécanisme de contrôle de congestion.
"Notification" de la gigue d'envoi due à la défaillance d'une route		
"Notification" de la gigue d'envoi due au changement de route		
"Notification" d'évitement de retransmission		Geler les transmissions et retransmissions pour la durée spécifiée dans le message. Réinitiatiser les temporisateurs.
"Notification" d'acquittement		Anticiper la transmission de nouvelles données s'il est établit par le DSR que le destinataire est à portée directe.
"Notification" explicite de congestion	Ctrl congestion	Invoquer le mécanisme de contrôle de congestion.
"Notification" de la baisse significative du niveau d'énergie	Contrôle données transférées	Modifier la fréquence de retransmission et/ou le débit de transmission.
"Mise à disposition" du taux de perte de paquet		Réajuster la fréquence de retransmission et le débit de transmission suivant une valeur élevée de ces paramètres établie à partir de seuils (indiquent l'état du canal).
"Mise à disposition" du SNR		
"Mise à disposition" du taux d'erreur bit BER		
"Mise à disposition" du RSS		Utiliser l'ACK de la couche liaison si le seuil de ce paramètre indique que le nœud en vis à vis est à portée directe
"Mise à disposition" du niveau d'énergie		Modifier la fréquence de retransmission et/ou le débit de transmission suivant les valeurs de ce paramètre établies à partir de seuils.
Service FEC "Activable"		Désactiver le mécanisme de vérification des données reçues (vérification du checksum).
Service ARQ "Activable"	Correction d'erreurs	Désactiver la fonction de correction d'erreur données s'il est établit par le DSR que le destinataire est à portée directe.

Table II.6. Tableau de description des interactions du protocole TCP

II.3.6.2. Cas du protocole DSR

Pour expliciter l'utilisation de chaque AACL faite par les fonctions du protocole DSR ainsi que les modifications du code source de ces fonctions que nous proposons, nous présentons ci dessous le tableau de description des interactions dudit protocole.

AACL	Fonction DSR	Exploitation au niveau du DSR
"Notification" d'évitement de retransmission	Contrôle de transmission	Geler les transmissions et retransmissions pour la durée spécifiée dans le message. Réinitiatiser les temporisateurs.
"Notification" d'acquittement		Prendre en compte la notification d'acquittement d'un paquet de la couche liaison (exploitation des intervalles SIFS).
"Notification" de la baisse significative du niveau d'énergie	Découverte de route + Contrôle de transmission + Récupération + Segmentation	Modifier la fréquence de retransmission de la fonction de contrôle de transmission, Modifier la fréquence de découverte de route, Désactiver la fonction de récupération, Désactiver la fonction de segmentation.
"Mise à disposition" du niveau d'énergie		Suivant les valeurs de ce paramètre établies à partir de seuils, Modifier la fréquence de retransmission de la fonction de contrôle de transmission, Modifier la fréquence de découverte de route, Désactiver la fonction de récupération, Désactiver la fonction de segmentation.
"Notification" pour la récupération d'un paquet	Récupération	Activer la fonction de récupération (qui vérifiera l'existence d'une nouvelle route).
"Notification" de la gigue d'envoi due au changement de route		Sur réception d'un message de récupération d'un paquet, notifier la prorogation du temps au TCP.
"Notification" de la puissance du signal reçu à partir d'un nœud	Routage	Enregistrer la puissance du signal reçue à partir d'un nœud comme métrique de sélection de route dans la table de route (indique les nœuds à portée directe).
"Notification" de la gigue d'envoi due à la défaillance d'une route	Gestion d'erreur de route	Sur réception d'un message d'erreur de route avec retransmission via un autre chemin ou lancement de la découverte de route, notifier la prorogation du temps à TCP.

Table II.7. Tableau de description des interactions du protocole DSR

II.3.6.3. Cas du protocole IP

Conformément au mécanisme précédemment utilisé, nous proposons ci–dessous le tableau de description des interactions du protocole IP. Ce tableau indique l'utilisation qui peut être faite de chaque AACL par les fonctions du protocole IP. Le tableau de description des interactions indique également la modification du code source de ces fonctions que nous envisageons.

AACL	Fonction IP	Exploitation au niveau d'IP
"Notification" explicite de congestion	Routage	Sur constat préventif de congestion, positionner le bit ECN dans l'entête TCP et acheminer le segment (l'entité TCP destinataire renverra ce renseignement à l'émetteur).
"Notification" de la baisse significative du niveau d'énergie	Routage, Segmentation	Désactiver la fonction de routage et de segmentation, se fier au routage DSR.
Service RSVP "Activable" de contrainte de délai	Routage	Sur réception des paramètres applicatifs RSVP de contraintes de délai, utiliser les requêtes–réponses de RSVP pour renvoyer une réponse explicite à l'application.
Service IntServ "Activable"		Sur réception des paramètres applicatifs IntServ de contraintes de délai, utiliser les requêtes–réponses de IntServ pour renvoyer une réponse explicite à l'application.
Service DiffServ "Activable"		Sur réception des paramètres applicatifs d'utilisation de DiffServ, utiliser le contrôle d'agrégat de trafic de DiffServ et renvoyer une réponse explicite à l'application.
"Mise à disposition" du niveau d'énergie	Routage, Segmentation	Suivant les valeurs de ce paramètre établies à partir de seuils, désactiver la fonction de routage et de segmentation, se fier au routage DSR.

Table II.8. Tableau de description des interactions du protocole IP.

II.3.6.4. Cas du protocole 802.11/couche liaison

Le mécanisme d'inférence précédemment utilisé nous permet également de proposer le tableau de description des interactions du protocole 802.11 de la couche liaison. Ce tableau fait référence à l'utilisation qui est faite de chaque AACL par les fonctions du 802.11 de la couche liaison ainsi que les modifications de leur code source qui sont envisagées.

AACL	Fonction 802.11	Exploitation au niveau du 802.11
"Notification" de la gigue d'envoi des paquets	Contrôle	Etablir l'acquittement global du paquet ou le décalage de son envoi. Si le décalage est établi, notifier l'événement aux couches concernées.
"Notification" d'évitement de retransmission		Sur constat de "mauvais" état du canal, de charge élevée du système, ou autre événement contraignant, notifier aux couches concernées la suspension d'envoi de paquets pendant une période spécifique.
"Notification" d'acquittement		Etablir l'acquittement global du paquet et notifier l'événement aux couches concernées.
"Notification" de la baisse significative du niveau d'énergie		Réajuster la fréquence de retransmission.
"Notification" pour la récupération d'un paquet		Sur établissement de l'inaccessibilité du prochain nœud (par interprétation du RSS, absence d'acquittement), solliciter la fonction de récupération du DSR.
"Notification" de la puissance du signal reçu à partir d'un nœud		Conserver la répartition d'un paquet en trame(s), conserver la valeur du RSS pour chaque trame acquittée, établir la moyenne des RSS du nœud et le notifier au DSR.
"Mise à disposition" du taux de perte de paquet		A intervalle régulier, établir et mettre à jour les valeurs des paramètres : Nombre de paquets non acquittés / nombre total de paquets envoyés dans la période Nombre de paquets reçus endommagés / nombre total de paquets reçus dans la période.
"Mise à disposition" du SNR		Utiliser ce paramètre de la couche physique pour établir la perte d'un paquet, la suspension de retransmission. Mettre sa valeur à la disposition des autres couches.
"Mise à disposition" du RSS		Etablir la valeur du RSS du nœud et la mettre à la disposition des autres couches.
"Mise à disposition" du taux d'erreur bit BER		Utiliser ce paramètre de la couche physique pour établir la perte d'un paquet, la suspension de retransmission.
"Mise à disposition" du niveau d'énergie		Suivant les valeurs de ce paramètre établies à partir de seuils, modifier la fréquence de retransmission.
Service VMAC "Activable"		Positionner l'indicateur d'activation, mettre à jour régulièrement les estimations locales de délais, de gigues, de collisions et de pertes de paquets
Service FEC "Activable"		Sur utilisation du FEC, positionner l'indicateur d'activation de ce mécanisme destiné à surmonter les pertes et la corruption des bits des paquets.
Service ARQ "Activable"		Sur utilisation d'ARQ, positionner l'indicateur d'activation de ce mécanisme destiné à offrir une fiabilité de transmission.

Table II.9. Tableau de description des interactions du protocole 802.11 couche liaison

II.4. Conclusion

La conception cross–layer est une nécessité pour les réseaux ad–hoc parce qu'elle permet d'améliorer leurs performances comparativement aux réseaux câblés qui ne sont pas handicapés par les défaillances de même nature. Il est important que cette conception se fasse dans un cadre standard pour favoriser l'évolution des modèles conceptuels des interactions entre les protocoles chaque fois qu'il faudra prendre en compte de nouvelles interactions, tout comme la mise en place de nouveaux modèles faisant intervenir d'autres protocoles justifie cette importance. L'importance de la méthode de conception se traduit également par la nécessité de conserver les acquis de l'architecture dont par exemple, la conception modulaire, la définition systématique des interactions entre les composants, la poursuite des objectifs à long terme quant à l'utilisation des réseaux, etc.

Dans ce chapitre de conception préalable à la mise en place des systèmes cross–layer, notre travail a consisté à formaliser les démarches à suivre à travers la méthode de conception RCL que nous proposons pour obtenir des modèles conceptuels cross–layer. L'application faite de la méthode de conception qui comportent sept étapes nous a permis de produire et de proposer des modèles d'interactions et des tableaux descriptifs des interactions.

La première étape de la méthode est une étape qui permet de fixer la pile de protocoles dont la mise en place des mécanismes d'optimisation cross–layer est envisagée.

L'étape 2 de recensement des AACL peut par analogie être comparée au procédé de remue-méninges ou brainstorming utilisé en industrie dans le cycle de vie d'un produit à améliorer pour identifier les insatisfactions du produit. Cette deuxième étape de la méthode RCL vise à favoriser l'émergence des idées pouvant engendrer des interactions cross–layer pour améliorer la performance du système qui doit fonctionner sur la base de la pile de protocole considérée à l'étape 1. Cette étape complète l'architecture considérée à l'étape 1 en lui adjoignant les différents services réseaux potentiellement utilisables par les protocoles considérés, tout comme elle favorise la prise en compte des paramètres significatifs disponibles sur une couche donnée, ainsi que des événements significatifs qui y surviennent et qui peuvent être exploités positivement par les autres couches. C'est pourquoi nous avons défini dans cette étape les actions atomiques cross–layer, c'est–à–dire les interactions pouvant avoir lieu sur la base de l'exploitation par d'autres couches d'un service, d'un paramètre ou d'un événement donné appartenant ou qui survient dans une autre couche.

L'étape 3 consacrée à la définition du tableau d'interaction des protocoles donne la distribution des interactions entre les protocoles et définit le(s) protocole(s) source(s), le(s) protocole(s) destination(s) de chaque interaction identifiée comme action atomique cross–layer à l'étape 2.

De même, l'étape 4 relative à la définition du tableau d'interaction des fonctions donne la distribution des interactions entre les fonctions. Cette étape définit la (les) fonction(s) source(s), la (les) fonctions destination(s) de chaque action atomique cross–layer identifiée. Les étapes 3 et 4 donnent des orientations relatives au travail modulaire qui doit être fait pour implanter les AACL.

A l'étape 5 de la méthode RCL, les modèles d'interaction des AACL déduits par catégorie d'AACL donnent un aperçu de la complexité des échanges créés par ces interactions cross–layer. Ces modèles regroupent la pile de protocoles considérée à l'étape 1, les sous–systèmes qui interviennent (y compris de façon implicite) ainsi que les modules systèmes à considérer. L'étape 5 de la méthode RCL permet de juger du compromis qu'il y a lieu de faire entre les interactions cross–layer et la nécessité de conserver les acquis de l'architecture.

L'étape 6 avec les tableaux de description des interactions par protocole décrit concrètement l'usage qui est fait de chaque AACL par les fonctions des protocoles considérés. Cette étape donne l'utilisation et l'exploitation qui sera faite par chacune des fonctions des différents protocoles lorsque les AACL seront implantées. Elle donne donc le travail de conversion à réaliser et constitue l'amorce de la transition entre l'aspect conceptuel et théorique de l'étude des AACL et la phase pratique de mise en œuvre qui doit déboucher sur la mesure des gains de performance obtenus.

La septième étape de la méthode est une phase de standardisation de l'implantation des AACL et permet de prendre en compte les mécanismes de communication du modèle global déjà arrêtés.

La prochaine phase de notre travail va consister à implanter ces modèles d'interactions cross–layer dans l'environnement ad–hoc en modifiant les codes sources des protocoles que nous avons choisis pour quantifier les gains de performances obtenus. Cette mise en œuvre passera par le choix des AACL à implanter et la proposition de mécanismes explicites novateurs qui doivent caractériser le fonctionnement des protocoles. C'est pourquoi dans les prochains chapitres, nous nous appesantirons sur l'étude de l'impact de l'état du canal sur les protocoles fiables de la couche transport, ceci parce que la nature variable et imprévisible de l'état du canal sans fil (avec également la mobilité qu'il offre) est la principale différence qui existe avec l'environnement filaire et constitue le handicap majeur pour lequel l'amélioration des performances des réseaux sans fil devient une nécessité au fur et à mesure de leur déploiement.

Chapitre III. Influence de l'état du canal dans la gestion des retransmissions de TCP

III.1. Introduction

Les modèles cross–layer permettent aux systèmes fonctionnant dans l'environnement des réseaux ad–hoc, d'intégrer des interactions efficaces entre les protocoles de différentes couches afin d'améliorer leurs performances, ceci du fait de leur soumission aux aléas du comportement dynamique. L'adaptation au fur et à mesure de l'évolution de l'environnement sans fil est le principe moteur sur lequel repose l'utilisation de ces interactions. Divers paramètres des interactions identifiées au chapitre précédent traduisent l'état du canal tels que par exemple la valeur du SNR, du BER, du taux de retransmission ou du taux de perte de paquets.
La congestion est un exemple d'état du canal à la fois filaire et sans fil. Le protocole TCP s'adapte à l'état de congestion des réseaux filaires au moyen du mécanisme de contrôle de congestion. Ce mécanisme peut être amélioré grâce à l'utilisation du bit ECN [RAM01].

Au chapitre précédent, nous avons proposé la méthode RCL destinée à faciliter la conversion des protocoles du réseau filaire vers l'environnement sans fil en utilisant le cross–layer. L'application de la méthode RCL à la pile de protocoles du réseau ad–hoc sélectionnée à l'étape 1 a donné les différents modèles conceptuels qui décrivent les échanges cross–layer entre les différentes couches. Le modèle représenté par le sous–système environnement, accessible en lecture/écriture à toutes les couches, centralise les paramètres des Actions Atomiques Cross–Layer (AACL) recensées qui décrivent l'environnement sans fil. Par exemple, l'état du canal sans fil est reflété par les AACL de mise à disposition du taux de perte de paquet, de mise à disposition du SNR, de mise à disposition du BER. Le tableau de description des interactions produit à la sixième étape de la méthode RCL propose une adaptation possible du protocole TCP à la variation de l'état du canal, à travers la modification du taux de transmission et la fréquence de ses retransmissions.

La retransmission est une fonction du protocole TCP qui définit la fiabilité de sa transmission. A la demande de l'entité TCP distante ou lorsque l'émetteur TCP le juge nécessaire par expiration du temporisateur d'envoi, un segment non acquitté est retransmis dans le réseau et un temporisateur de retransmission est démarré. La politique traditionnelle de gestion de ce temporisateur de retransmission telle qu'appliquée par le protocole TCP admet qu'il acquiert une valeur croissante multiple de sa valeur précédente à chaque expiration sans réception d'acquittement. Cette politique ne prend pas en compte la variation de l'état du canal. La persistance d'un mauvais état engendre l'indisponibilité du lien sans fil et par conséquent des retransmissions par la couche transport sur expiration du temporisateur d'envoi.

Pour mettre en œuvre l'adaptation du protocole TCP à l'environnement sans fil préconisée au chapitre précédent, nous proposons dans ce chapitre une autre politique de gestion du temporisateur de retransmission. Cette politique que nous appelons "politique persistante de retransmission" repose sur l'exploitation des données cross–layer d'état du canal du sous–système environnement. Elle présente divers avantages dont l'évitement de saturation des files d'attente d'envoi d'un nœud mobile ou bien encore la minimisation de la latence d'envoi. La latence d'envoi d'un paquet provient du fait que, entre deux tentatives traditionnelles d'envoi, le délai d'attente avant retransmission peut être inutilement rallongé si le canal devient disponible très tôt après la dernière tentative. L'importance de ces gains escomptés de la nouvelle politique de retransmission de TCP se traduit dans l'amélioration de la qualité de la communication qui se mesure par une gigue et une latence aussi faibles que possible, un débit de transmission aussi grand que le permet le support de transmission sujet à des variations et l'assurance de la fiabilité de la transmission.

III.2. Fonctionnement du protocole TCP

La pile de protocoles TCP/IP du modèle en couches a assuré pendant une période assez longue, un transport efficace et fiable des données des applications du réseau Internet. Le contrôle de flux, le contrôle de congestion et l'utilisation des retransmissions sont les principaux mécanismes mis en œuvre par le protocole TCP pour assurer ce service fiable. L'émergence de nouveaux services laissant apparaître des insuffisances a conduit à la proposition de solutions pour améliorer les performances du protocole TCP. Le "Fast retransmit / Fast recovery" et le "New Reno" sont des exemples de ces solutions.

III.2.1. Le contrôle de flux

Le contrôle de flux est un mécanisme mis en œuvre par l'entité TCP émettrice pour éviter de déborder la mémoire du récepteur. Ce mécanisme implique la vérification préalable par l'émetteur de la disponibilité de l'espace de stockage au niveau du récepteur grâce à l'utilisation de la fenêtre d'émission SWND représentant le nombre de segments pouvant être transmis sans provoquer de débordement dans la file d'attente du récepteur. Les valeurs de SWND proviennent de l'échange permanent des tailles des fenêtres de contrôle de flux par les entités TCP en communication, à partir d'un champ réservé de l'entête du segment.

III.2.2. Le contrôle de congestion

Le contrôle de congestion assuré par le protocole TCP consiste à adapter les débits de transmission des sources émettrices en fonction de la charge du réseau. L'état de congestion du réseau est détecté par l'émetteur par interprétation des pertes supposées qui surviennent comme étant la conséquence de cette congestion. De ce fait, la source TCP diminue son débit de transmission pour éviter d'aggraver l'état de saturation en fournissant un trafic supplémentaire. Il faut distinguer trois états différents par lesquels passe une source TCP pour le contrôle de congestion, à savoir le démarrage lent (Slow Start), l'évitement de congestion (Congestion Avoidance) et le recouvrement rapide (Fast Recovery). Chaque état gère de façon appropriée la fenêtre de congestion CWND qui représente le nombre de segments (d'octets) pouvant être transmis sans risque de congestionner le réseau. Les changements d'état se font lorsque la source TCP détecte la perte de segment ou lorsque sa fenêtre CWND atteint un seuil prédéfini. Le changement d'état est donc de ce fait un mécanisme à seuil.

Le "Slow Start" et le "Congestion Avoidance" sont deux mécanismes proposés dans la version de base de TCP appelée "TCP Tahoe" [RFC0793]. Ils permettent de contrôler la fenêtre de congestion. Le mécanisme de "Fast retransmit / Fast recovery" est une amélioration proposée par la version "TCP Reno" pour détecter plus rapidement la perte de segment grâce à l'usage de trois acquittements dupliqués.

III.2.2.1. Le mécanisme de démarrage lent

La fenêtre de congestion CWND est initialisé au départ à 1 MSS (Maximum Segment Size), ce qui donne l'autorisation à la source d'émettre un segment. Lorsque ce segment émis est acquitté, la fenêtre CWND est doublée, ainsi de suite, ce qui donne une allure exponentielle à la courbe qui représente l'évolution de cette fenêtre de congestion.

III.2.2.2. Le mécanisme d'évitement de congestion

Lorsque la fenêtre de congestion CWND atteint un seuil "threshold", TCP s'impose un infléchissement et le rythme d'augmentation devient linéaire. Dans cette phase appelée d'évitement de congestion, la fenêtre CWND n'augmente pas de plus de 1MSS par temps aller retour RTT (Round Trip Time).

III.2.3. La version TCP Reno

Le trafic de paquets des réseaux IP est soumis à des modifications d'itinéraire, ou à des retransmissions, …, qui causent des déséquencements dans les flux ordonnés de segments TCP. Lorsqu'un récepteur TCP détecte les segments manquants dans un flux qui arrivent dans le mauvais ordre, il doit envoyer un accusé de réception stipulant le numéro du prochain segment attendu pour que l'émetteur retransmette les données sollicitées. Les accusés de réception peuvent être en double lorsqu'un accusé de réception porte le même numéro qu'un autre précédemment envoyé pour le dernier segment ordonné reçu.

Le mécanisme de " Fast Retransmit" permet au protocole TCP de gagner du temps dans le recouvrement des pertes de paquets. Par ce mécanisme, un accusé de réception reçu en trois exemplaires permet à l'émetteur de déduire la perte du segment indiqué et de le retransmettre sans attendre l'expiration du temporisateur de retransmission [RFC2581]. Après cette phase, l'émetteur réduit sa fenêtre de congestion de moitié et invoque l'algorithme de "Fast Recovery" qui appelle la phase d'évitement de congestion à partir de cette fenêtre réduite. Le protocole n'a pas recours à la phase de "slow start" comme dans le cas de l'expiration du temporisateur.

III.2.4. La version TCP NewReno

Dans la version TCP Reno, l'algorithme de "Fast Recovery" prend fin lorsqu'il reçoit un acquittement supérieur à celui qui avait été dupliqué. Le document [RFC2582] propose la version NewReno qui est une évolution de la version Reno. Le mécanisme proposé consiste à rester en mode "Fast Recovery" tant que l'algorithme n'a pas reçu d'acquittement pour le segment envoyé portant le plus grand numéro de séquence, en d'autres termes, le mode "Fast Recovery" est conservé si l'acquittement reçu qui est supérieur à celui qui avait été dupliqué n'acquitte pas tout ce qui a été envoyé. Ce mécanisme permet de récupérer rapidement plusieurs segments perdus sans recours aux acquittements sélectifs (SACK) qui peuvent être implantés conjointement à la version NewReno.

III.2.5. La version TCP Vegas

Pour mesurer la portée possible de son volume de transmission, TCP provoque volontairement une congestion, du fait que dans la phase d'évitement de congestion, la taille de la fenêtre de congestion augmente systématiquement de 1 à chaque période RTT, avant que le protocole réduise son débit à un point de fonctionnement.

La version TCP Vegas [MAL94] permet de trouver le débit de transmission acceptable sans provoquer de congestion. Dans cette version, l'algorithme d'évitement de congestion est remplacé par Vegas qui permet à la fenêtre de congestion d'augmenter d'une unité pendant le temps RTT, de rester constante et aussi de diminuer au besoin, par simple interprétation de la valeur du RTT. En effet, lorsque le réseau est proche de la congestion, les files d'attente des routeurs sont pleines et engendre une augmentation du RTT, tandis qu'au contraire, la diminution du RTT indique un réseau dégagé. Pour le cas du RTT long, la fenêtre de congestion diminue, et au contraire, le RTT court permet d'augmenter sa taille. En cas de perte de segment, la fenêtre n'est pas diminuée de moitié mais de ¾.

III.2.6. Les acquittements sélectifs

Les acquittements sélectifs (SACK) peuvent être incorporés au protocole TCP pour permettre de pallier la perte de plusieurs segments par fenêtre de congestion sans avoir recours à un ou plusieurs aller–retour par perte [RFC2018]. Le document [RFC2883] propose une extension des accusés de réception sélectifs (SACK) pour permettre un fonctionnement plus robuste dans un environnement de paquets réordonnés ou répliqués, dans un environnement de perte d'accusé de réception et/ou de temporisations de retransmission anticipées.

III.2.7. Mise en évidence de la temporisation traditionnelle de TCP

Nous mettons en œuvre dans l'environnement ns-2 [FAL03] un scénario de simulation simple composé de deux nœuds qui communiquent à travers un canal sans fil. La couche application du nœud source émet un trafic CBR et utilise le protocole TCP au niveau transport. La couche transport du nœud de destination enregistre les données qu'elle reçoit toutes les ½ seconde. Le canal retour est un canal absorbant, de telle sorte qu'aucun acquittement n'est reçu au niveau du nœud émetteur. Ceci permet de mettre en évidence l'évolution de la valeur du temporisateur d'attente d'envoi avant retransmission. Cette évolution est représentée par la courbe de la figure III.1 qui montre la fréquence de l'envoi du paquet d'établissement de connexion ainsi que l'augmentation des intervalles de retransmission.

Le temps initial d'expiration du temporisateur appelé ITO (Initial Time Out) représente le temps au bout duquel un paquet est retransmis lorsque son acquittement n'est pas reçu par le TCP émetteur. Ce temps ITO est fixé à 3 secondes [RFC1122] et le temps de la fin d'une connexion TCP est en général fixé à 3 minutes. Cette modélisation entraîne nécessairement un trafic supplémentaire dû aux requêtes envoyées pour solliciter la retransmission du premier paquet qui a été retardé dans la file d'attente et pour lequel un acquittement n'a pas été reçu.

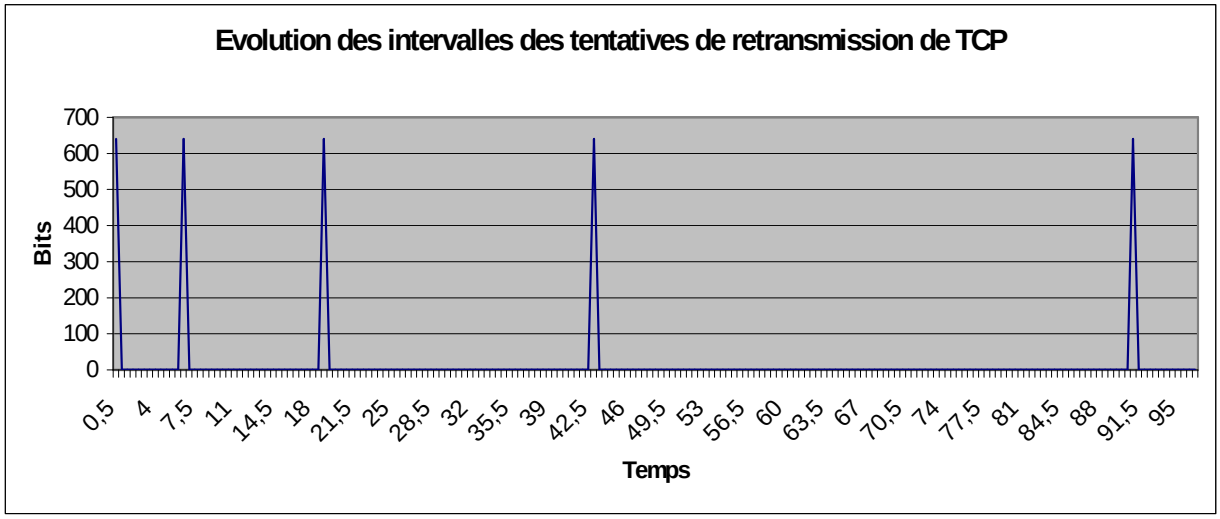


Figure III.1. Evolution de la valeur du temporisateur d'attente avant retransmission

III.3. Principe de la politique persistante

Comme précédemment énoncé, une faible gigue, une grande bande passante, une faible latence sont des éléments supplémentaires d'appréciation de la qualité d'une communication. Au niveau de la couche transport, TCP et ses variantes fournissent des connexions fiables en adoptant le principe de retransmission des segments. Ce qui constitue un facteur de latence. Il est de ce fait important qu'elle engendre une latence supplémentaire minimale. C'est pourquoi nous proposons un schéma interne à la politique de retransmission dans le but d'améliorer la latence d'envoi.

L'idée de base de la temporisation persistante provient de la classification des mécanismes de transmission au niveau MAC reposant sur l'écoute de la porteuse (carrier sens). Les protocoles persistants sont distingués des non persistants. Dans les deux cas, le terminal émet lorsque le canal est libre. Quand le canal est occupé, un protocole non–persistant impose au terminal prêt à émettre un délai d'attente aléatoire pour effectuer une nouvelle tentative alors qu'un protocole persistant continue d'écouter le canal et émet dès que la porteuse est libre. Le protocole persistant réduit les périodes d'inactivité dues aux délais d'attente, même si, au niveau physique, il augmente fortement le risque de collision en fonction de la durée d'émission des messages. Ce risque de collision n'existe pas à la sortie de la couche transport à laquelle notre schéma applique ce même principe pour la sortie des paquets ré-émis.

De façon plus générale, la persistance peut être exprimée en terme de temps ou de nombre maximum de retransmissions. Le schéma de temporisation que nous proposons utilise l'expression de la persistance en terme de temps.

La politique traditionnelle de gestion de la temporisation de TCP (le back–off exponentiel) est apparentée au mécanisme du protocole non persistant décrit ci–dessus. L'absence d'acquittement d'un segment envoyé par le protocole TCP à la couche transport est détectée par l'expiration d'un temporisateur. Le segment est retransmis et le temporisateur mis à jour avec une nouvelle valeur. Cette nouvelle valeur s'accroît à chaque expiration d'un multiple de sa valeur précédente. Chaque nouvelle valeur de ce temporisateur détermine la durée d'attente avant la prochaine re-émission. Ce principe d'accroissement exponentiel du délai d'attente avant retransmission permet d'éviter un délai trop court qui aura comme inconvénient majeur d'engendrer de nombreuses retransmissions. Mais ce mécanisme global de gestion des retransmissions ne distingue pas les prolongements de délais dus aux facteurs propres à l'environnement sans fil. Ces facteurs peuvent être par exemple le cas de la mobilité des nœuds qui engendre une modification de la route source d'un paquet DSR lors de sa transmission, tout comme une erreur de route qui occasionne la relance de la découverte de route de DSR, ou tout simplement un mauvais état du canal qui bloque temporairement les émissions du nœud mobile. Si ces facteurs sont la cause du prolongement du délai de transmission (du fait des retransmissions), une forte latence sera observée lorsque la transmission devient à nouveau possible après la dernière tentative d'envoi et avant la prochaine tentative, alors que dans ces conditions, il n'est normalement plus question d'attendre l'expiration du temporisateur pour une nouvelle retransmission.

En exploitant l'information cross–layer relative à l'état du canal donnée par le sous–système environnement du modèle conceptuel, une autre politique de gestion du temporisateur de retransmission de TCP peut être mise en place.

Le principe de la temporisation persistante consiste pour TCP, à la première expiration du temporisateur d'attente d'acquittement, à rechercher l'explication appropriée dans les paramètres du sous–système environnement. Lorsque la cause provient d'un mauvais état du canal, à la place de l'algorithme traditionnel qui octroie une valeur croissante au temporisateur d'envoi, le protocole TCP va adopter un comportement persistant en observant la prochaine modification favorable de l'état du canal dans le sous–système environnement, pendant toute la durée de vie du segment en instance de retransmission à la couche transport. Si ce segment est ré–émis à l'instant où l'état du canal le permet, la durée d'attente peut ainsi être écourtée, ce qui a pour effet intuitif d'engendrer un gain de temps relatif à l'évitement de la latence d'envoi.

Dans ce schéma de temporisation persistante, l'abandon de la transmission ne sera pas déclenchée par l'expiration du nombre maximal de retransmission autorisée à la couche transport, mais plutôt par l'expiration de la durée de vie ordinaire du paquet à la couche transport.

La politique persistante de temporisation sera également plus efficace lorsqu'elle est utilisée à la place de la politique traditionnelle dans les cas des AACL de mise à disposition de la gigue d'envoi due à une erreur de route et de la gigue d'envoi due à un changement de route du protocole de routage DSR.

Un autre apport de la politique persistante de temporisation se rapporte au calcul du nombre de tentatives infructueuses de retransmission de segments. A titre de preuve intuitive, le calcul du nombre de tentatives infructueuses en fonction du nombre brut de segments envoyés par la couche transport, avec ou sans distinction de leur identifiant, démontre que la politique traditionnelle non persistante de retransmission aura un inconvénient majeur, dans les cas énoncés ci–dessus, du fait qu'elle engendre un nombre de tentatives plus élevé que celui de la politique persistante qui aura comme effet de le stabiliser. L'importance du calcul du nombre de tentatives infructueuses se traduit par le fait que les retransmissions sont synonymes de consommation non négligeable et même considérable d'énergie, alors que la durée de vie et l'autonomie des nœuds sans fil dépend fortement des mécanismes de sauvegarde et d'évitement de perte par gaspillage de cette énergie.

III.4. Généralités sur les critères d'évaluation des performances

Il est possible de définir un modèle de performance de réseau sur la base de cinq métriques principales qui sont le débit, le délai, la variation de délai, le taux de perte, le taux d'erreur. Le document [RFC2330] définit des métriques de performance d'IP.

Le débit ou bande passante entre deux systèmes communicants est défini comme étant le taux de transfert maximum pouvant être maintenu entre deux points c'est à dire le nombre de bits que le réseau est capable d'accepter ou de délivrer par unité de temps.

Le délai de propagation dépend de la nature du support physique utilisé. Le temps d'émission est fonction du débit binaire et de la taille des paquets. Le temps de traitement dépend des équipement intermédiaires utilisés dans la communication comme par exemple la traversée d'un codeur/décodeur. Le délai cumulé dans les files d'attente des routeurs dépend des tailles des tampons mémoire et de l'encombrement, par exemple, un délai de file de 50ms sur un lien à 25Mbps indique qu'il y a environ 15 koctets en attente dans les routeurs.

Le délai de transit de bout en bout est le temps écoulé entre l'envoi d'un paquet par un émetteur et sa réception par le destinataire. C'est la somme du délai de propagation, du temps d'émission, du temps de traitement, du délai cumulé dans les files d'attente des routeurs, du délai introduit par le tampon de compensation de la gigue pour assurer la synchronisation.

Dans l'évaluation de performance relative au modèle de comparaison de la temporisation persistante avec la temporisation traditionnelle de TCP, nous considérons la gigue de transmission des paquets comme étant le temps qui s'écoule entre la disponibilité du support de transmission et l'envoi effectif des paquets. L'instant de l'envoi effectif de la temporisation persistante est proche de l'instant de disponibilité du lien sans fil tandis que celui de la temporisation traditionnelle est associé à l'expiration du temporisateur de retransmission. De façon générale, la gigue est considérée comme la variation de délai de bout en bout. La gigue du délai pour un flux de paquets est définie comme étant la différence maximum de délai entre les paquets du flux pris deux à deux. L'IETF a défini la notion de gigue instantanée comme étant la variation instantanée (instantaneous packet delay variation : IPDV), c'est à dire la différence du délai de transmission entre deux paquets k et k+1 consécutifs. Elle reflète l'évolution de l'état de congestion du lien ou du chemin. Si elle est stable, la charge du lien ou du chemin est constante. Si elle augmente, elle indique une dérive vers un état de congestion. Elle est utilisée par TCP Vegas pour anticiper les pertes de paquets, donc les congestions et mettre en œuvre les mécanismes d'évitement de congestion plus rapidement. La gigue ne permet d'anticiper un état de congestion que si le délai de bout en bout est relativement grand.

Le taux de perte correspond au rapport entre le nombre de paquets non arrivés sur le nombre total de paquets transmis.

Le taux d'erreur correspond au nombre de bits reçus erronés sur le nombre total de bits reçus ou le nombre total de paquets erronés sur le nombre total de paquets reçus.

III.5. Modélisation Algorithmique et Mathématique

Le but de la politique persistante de contrôle des retransmissions est de maintenir le débit de sortie au niveau le plus élevé possible. Pour montrer l'efficacité de notre schéma dans l'amélioration de la latence d'envoi et procéder au calcul du gain obtenu à l'aide de la politique persistante de retransmission par rapport à la politique traditionnelle de TCP, nous utilisons dans cette partie, un modèle hybride algorithmique avec des notations mathématiques. Nous appelons latence maximale, le temps maximal théorique qui peut séparer chacune des ré–émissions des deux politiques de retransmission après blocage temporaire du canal. De même, nous mettons en exergue les facteurs d'étirement de cette latence qui vont rallonger le temps d'attente avant retransmission.

Les schémas algorithmiques que nous présentons ne sont pas forcement optimaux. Ils ne sont utilisés que pour permettre de mieux appréhender le comportement des deux mécanismes de temporisation dans le temps. Ces schémas algorithmiques ne tiennent pas aussi compte de l'aspect événementiel de l'implantation des protocoles, ni des mécanismes globaux du protocole TCP.
Considérons :

- Pk (pour Packet) le segment TCP à envoyer,
- Tr (pout Timer) le temporisateur d'attente avant retransmission,
- Ts (pour Start Time) le temps de démarrage de l'envoi du paquet.

Le principe de la politique persistante de retransmission impose au protocole 802.11 de la couche liaison une fréquence de mise à jour des variables d'état du canal du sous–système environnement. Elle impose au protocole TCP une fréquence continuelle d'observation de l'état du canal pour détecter sa variation. Ces fréquences de mise à jour et d'observation ont donc une importance notoire dans la mise en œuvre de la retransmission persistante par utilisation du cross–layer. C'est pourquoi, nous consacrons le point suivant à l'étude de la fréquence d'observation de TCP, puisque celle du 802.11 est intimement liée à la nature imprévisible du canal. Il est normalement supposé que la mise à jour des variables d'état du canal du sous–système environnement n'a lieu que lorsque l'état courant du canal change par rapport à son état précédent. De ce fait, lorsque le protocole calcule les nouvelles valeurs de ces paramètres, la mise à jour n'a lieu que lorsque les valeurs calculées sont différentes des dernières valeurs conservées. Par contre, la date de ce calcul est un paramètre important qui sera pris en compte dans la détermination de la continuité de l'évaluation.

III.5.1. Détermination de la fréquence d'observation de l'état du canal du protocole TCP

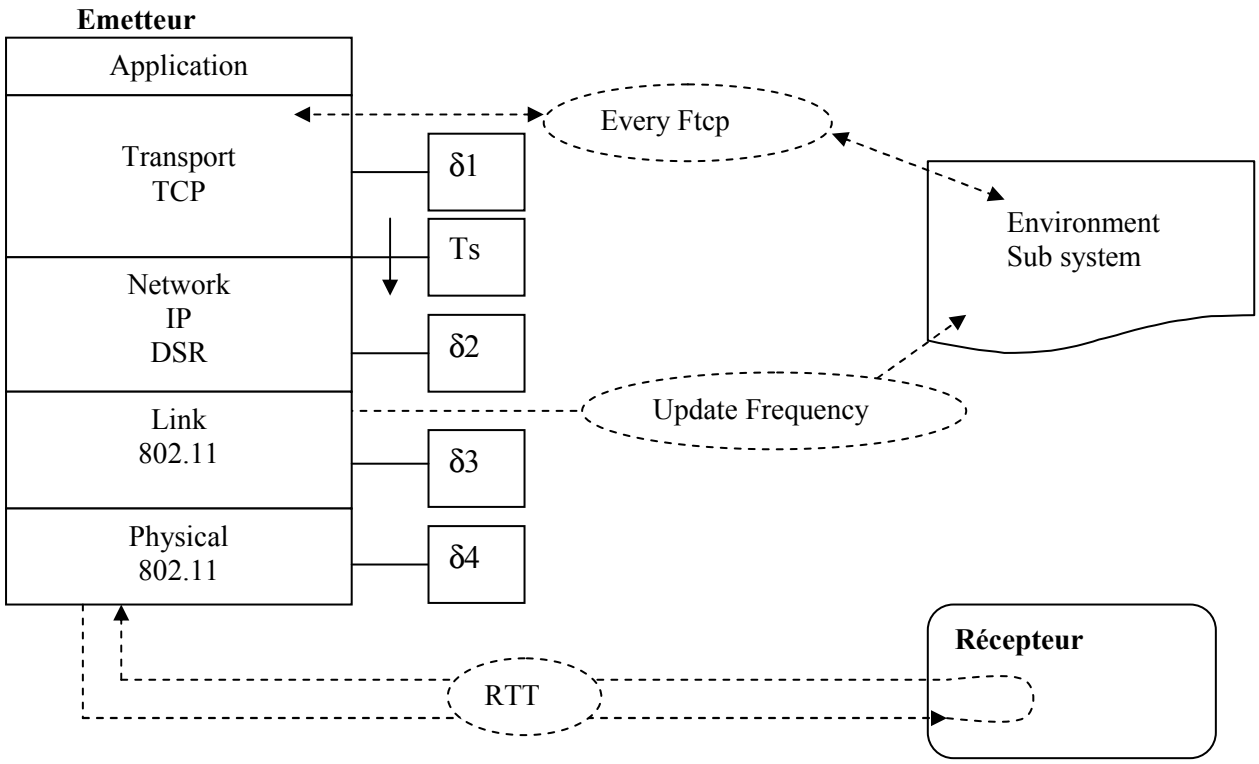


Figure III.2. Temps à prendre en compte dans la politique persistante de temporisation.

Légende
Ftcp : fréquence TCP d'observation des variables d'état du canal du sous–système environnement.
δ1 : durée maximale de mise en zone tampon du paquet à la couche transport avant sa suppression.
δ2 : durée maximale de mise en zone tampon du paquet à la couche réseau avant sa suppression.
δ3 : durée maximale de mise en zone tampon du paquet à la couche liaison avant sa suppression.
δ4 : durée maximale de mise en zone tampon du paquet à la couche physique avant sa suppression.

Les durées maximales ci–dessus sont relatives aux politiques de gestion des tampons d'attente d'envoi de paquets au niveau de chaque couche. L'exemple du mécanisme de découverte de route du DSR illustre l'utilisation de ces durées maximales en zone tampon pour la gestion d'un paquet en cours d'envoi. Lorsque le paquet de requête de route est envoyé à travers le réseau, le paquet original est stocké dans le tampon appelé "Send Buffer". Ce tampon contient la date d'insertion de chaque paquet en attente. Les paquets sont supprimés du tampon après une période délimitée par un temporisateur. Mais cette durée de vie peut être écourtée pour éviter un débordement du tampon. La politique de sortie FIFO est ainsi appliquée ou toute autre stratégie de suppression des paquets les plus anciens même si le temporisateur de conservation n'a pas expiré.

III.5.1.1. Modélisation mathématique

La figure III.1 présente la pile complète de protocoles d'un nœud émetteur accompagnée de divers paramètres dont les temps de présence d'une PDU (Protocol Data Unit) donnée dans chaque couche ($\delta 1$, $\delta 2$, $\delta 3$, $\delta 4$). La figure comporte également l'indication de la fréquence *Ftcp* d'observation de l'état du canal, l'indication du RTT et celle de la fréquence de mise à jour de l'état du canal par la couche liaison. Nous utilisons le terme PDU dans la mesure où les données vont être traitées par les différentes couches.

Du fait qu'une boucle de ré–émission TCP est plus grande qu'une boucle DSR, elle même supérieure à une boucle 802.11, nous déduisons les inégalités suivantes :

$$\delta 1 > \delta 2 > \delta 3 > \delta 4 \quad (1)$$

Après sa sortie de la couche transport, le temps maximal qu'une PDU pourrait passer dans les couches inférieures avant sa suppression, que nous appellerons ILT (pour Inferior Living Time), est donné par :

$$ILT = \max(\delta 2, \delta 3, \delta 4) = \delta 2 \quad (2)$$

Nous déterminons la fréquence d'observation Ftcp en respectant la contrainte qui permet d'éviter une fréquence trop grande, auquel cas le gain en latence s'amenuisera, de même qu'une fréquence trop petite alourdira la charge du système, étant entendu que toute modification significative de la valeur de l'état du canal sera notifiée au protocole TCP par message interne.
Si l'état du canal le permet, une retransmission TCP n'est opportune que si la durée de vie de la PDU aux niveaux inférieurs a expiré. De ce fait, nous alignerons la Fréquence d'observation Ftcp sur le temps maximal de mise en zone tampon de la PDU dans les couches inférieures de la façon suivante :

$$Ftcp = ILT = \delta 2 \quad (3)$$

III.5.1.2. Principe de Notification Explicite de Changement favorable d'Etat du canal (NE-CE)

Par cette équation (3), la ré–émission d'un paquet par TCP n'est probable qu'après expiration de la durée de vie de la PDU aux niveaux inférieurs. La re–émission dépendra à la fois de cette durée de vie et de l'état du canal. Bien que la re-émission traditionnelle de TCP soit contrôlée par l'expiration du temporisateur de retransmission, des messages provenant des couches inférieures sont déjà implantés pour informer TCP de la suppression de son segment. Le principe de ré–émission probable cadrée sur la durée de vie du paquet aux niveaux inférieurs peut donc facilement exploiter ces messages de suppression des segments TCP par les couches inférieures. Cette exploitation se fera de façon complémentaire au principe d'attribution de la valeur de l'ILT à la fréquence *Ftcp* d'observation de l'état du canal qui déterminera la probabilité de retransmission.

Un mécanisme complémentaire à l'exploitation des messages de suppression de segments sera mis en œuvre. Il utilise des messages qui notifient explicitement le changement favorable de l'état du canal au protocole TCP. Cette notification explicite de changement favorable d'état du canal (NE–CE) peut être implantée d'au moins deux façons. La première consiste à mettre en place une communication directe via le sous–système interface entre le

protocole 802.11 de la couche liaison et le protocole TCP de la couche transport. La deuxième implantation possible des messages NE–CE consiste à utiliser un module d'interprétation des valeurs des paramètres d'état du canal greffé au sous–système environnement. Ce module aura la fonction d'envoyer ces messages explicites au protocole TCP.

III.5.1.3. Modélisation hybride

Nous présentons ici le comportement temporel des deux politiques de retransmission. Pour modéliser le déroulement algorithmique de ces deux politiques, nous faisons d'abord des hypothèses sur les primitives algorithmiques destinées à simplifier la présentation de leur comportement. Nous présentons ensuite le déroulement algorithmique de la politique traditionnelle et le squelette du modèle algorithmique qui la représente et enfin, nous présentons le squelette du modèle algorithmique de la politique persistante.

Considérons :
- La primitive *Emettre(Pk)* représente la fonction d'envoi du segment de TCP.
- La primitive *ReEmettre(Pk)* représentant la fonction de retransmission. Nous attribuons à cette fonction le rôle de vérifier qu'un ACK (acquittement) du segment à ré–émettre n'a pas été reçu entre–temps avant de procéder à la ré–émission proprement dite.
- La primitive *Observer(EtatCanal, Tr)* est une fonction bloquante chargée de vérifier l'état du canal avec une périodicité *Ftcp*. La sortie du blocage est conditionnée par un meilleur état du canal ou l'expiration du délai δl de vie du segment au niveau transport.
- La primitive *Attendre(Temps)* représentant la fonction d'attente avant retransmission.
- La primitive *Expire(Temporisateur)* permet de vérifier qu'un temporisateur n'a pas atteint sa valeur limite.

Dans la politique traditionnelle, le temporisateur *Tr* d'attente avant retransmission prend une valeur multiple de sa valeur précédente à chaque expiration. Ceci ramène donc le temporisateur *Tr* à une valeur multiple de sa valeur initiale de départ.

La suite des valeurs X_i sera appelée les multiplicateurs de l'intervalle initial d'attente avant retransmission symbolisé par *Tr*.
Soit *NbRetransMax* le nombre maximal de retransmissions. Ce nombre maximal non précisé dans les RFC TCP est implanté dans la pratique.

La formule ci–dessous représente le calcul de l'intervalle initial d'attente avant retransmission fait par TCP :

$$Tr = \alpha \,.\, \text{estimation(RTT)}, \text{ avec généralement } \alpha = 2 \qquad (4)$$

Ce paramètre α permet d'éviter de retransmettre trop vite des segments. L'estimation du RTT se fait au travers du mécanisme de type moyenne mobile.

a) Modélisation algorithmique de la politique traditionnelle de retransmission

Squelette algorithmique de la politique traditionnelle :

```
Tr = RTT + estimations
i = 1
Emettre(Pk)
Attendre(Tr)
Tant que Pas d'ACK et i < NbRetransMax
        ReEmettre(Pk)
        Tr = Tr * X_i
        i = i + 1
        Attendre(Tr)
Fin tant que
```

b) Modélisation algorithmique de la politique persistante de retransmission

Pk_TCP_TTL est un temporisateur dont la valeur initiale désigne le délai maximal d'un paquet à la couche transport.

Squelette algorithmique :

```
Pk_TCP_TTL = δ1
Tr = RTT + estimations
Emettre(Pk)
Attendre(Tr)
Tant que Pas d'ACK et non Expire(Pk_TCP_TTL)
        Observer(EtatCanal, Pk_TCP_TTL)
        Si non (Expire(Pk_TCP_TTL)) et (EtatCanal == "OK")
                ReEmettre(Pk)
                Attendre(δ2)
        Fin si
Fin tant que
```

c) Mise à l'échelle temporelle

Pour faciliter la lisibilité de la mise à l'échelle temporelle, nous notons *E* la primitive *Emettre(Pk)* et *R* la primitive *ReEmettre(Pk)*. Nous supposons que l'état du canal bloque la sortie des paquets d'un nœud pendant une durée *a* représentant le temps après la i–ème tentative de retransmission de la politique traditionnelle. Dans les présentes simulations, nous avons pris *a* constant. Les ré–émissions ont lieu à toutes les tentatives dans la politique traditionnelle, elles n'ont lieu que lorsque l'état du canal le permet dans la politique persistante.

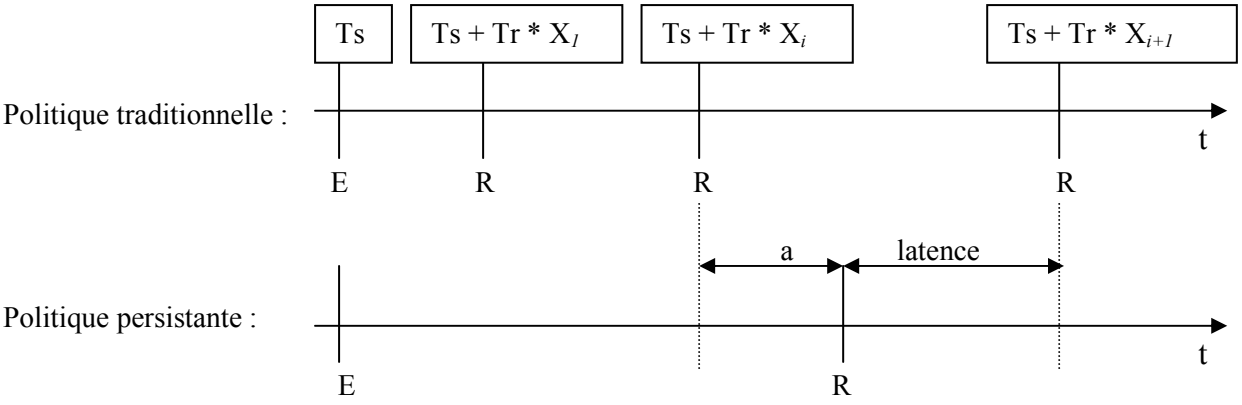


Figure III.3. évolution temporelle du comportement des deux politiques de retransmission

d) Calcul de la latence

La formule de calcul de la latence est donnée par :

$$\text{Latence} = \text{Tr}\,(X_{i+1} - X_i) - a \qquad (5)$$

Cette définition de la latence nous permet d'observer qu'elle diminue avec *a* et qu'elle augmente avec *i* en raison de l'augmentation exponentielle de la valeur de la temporisation dans la méthode traditionnelle. Nous appelons l'intervalle $X_{i+1} - X_i$ facteur d'étirement de la latence.

e) Calcul du taux de tentatives infructueuses (TTI)

Dans la politique traditionnelle de retransmission, tous les paquets ré-émis pendant l'impossibilité d'envoi engendrée par un mauvais état du canal sont perdus. Le nombre de tentatives infructueuses *Nt* (dont l'indice *t* désigne la politique traditionnelle) lorsque la transmission devient possible entre l'instant $Ts + Tr * X_i$ et l'instant $Ts + Tr * X_{i+1}$ est donné par :

$$\text{Nt} = i \text{ avec } i \geq 1 \qquad (6)$$

Puisque depuis le premier envoi, seul l'envoi qui suit le changement positif de l'état du canal a des chances d'être un succès, si un nouveau changement défavorable de l'état du canal n'a pas lieu avant l'expiration de la durée d'attente restante. Il pourrait être intéressant d'observer ce phénomène sous divers modèles de variation de l'état du canal. La mesure du gain obtenu par simulation se fait par fixation d'un simple modèle de variation de l'état du canal sans fil.

Dans la politique persistante de retransmission, seul le premier paquet émis pendant l'impossibilité d'envoi engendrée par un mauvais état du canal est perdu. Ainsi, le nombre de tentatives infructueuses *Np* (dont l'indice *p* désigne la politique persistante) lorsque la transmission devient possible entre l'instant $Ts + Tr * X_i$ et l'instant $Ts + Tr * X_{i+1}$ est invariant et est donné par :

$$\text{Np} = 1 \qquad (7)$$

En supposant que η segments ont été au total émis par TCP, le taux de tentatives infructueuses *TTIt* de la politique traditionnelle est donné (en pourcentage) par :

$$\text{TTIt} = (\text{Nt} * 100) / \eta \qquad (8)$$

Le taux de tentatives infructueuses *TTIp* de la politique persistante est donné (en pourcentage) par :

$$\text{TTIp} = (\text{Np} * 100) / \eta \qquad (9)$$

Du fait de (6) et (7), l'inégalité suivante est toujours vérifiée :

$$\text{TTIp} \leq \text{TTIt} \qquad (10)$$

f) Déduction de l'énergie perdue

L'importance du taux de tentatives infructueuses se mesure en terme d'énergie consommée par le mécanisme de retransmission, sachant que la gestion d'énergie est un facteur primordial qui définit la durée de vie d'un réseau ad–hoc.

Le caractère fini de la source d'énergie d'un nœud mobile du réseau sans fil est illustré dans les travaux pionniers de Gallager [GAL98]. L'auteur indique qu'un réseau ad–hoc avec des nœuds disposant d'une énergie finie dispose forcement d'un nombre fini de bits. Un nœud donné du réseau a donc un nombre fini de bits qu'il peut utiliser pour son fonctionnement avant d'épuiser son énergie. Soit E, l'énergie utilisé à la couche transport pour envoyer un segment.

Etant donné que l'allocation de l'énergie en terme de bits contenus dans un paquet tombe nécessairement sous la contrainte de l'optimisation pour que la durée de vie du nœud soit prolongée, nous choisissons de comparer les deux politiques de temporisation à travers l'énergie consommée par nombre de paquets émis à la couche transport pendant un mauvais état du canal.

Soit *Ep* l'énergie (associée aux paquets) consommée par la politique de retransmission persistante et *Et* celle de la politique traditionnelle. Les valeurs de *Ep* et *Et* sont données par :

$$Ep = Np \times E \quad (11)$$
$$Et = Nt \times E \quad (12)$$

L'influence de (6) et (7), se traduit toujours par :

$$Ep \leq Et \quad (13)$$

La politique persistante est moins consommatrice d'énergie lorsque le mauvais état du canal se prolonge, du fait que la politique traditionnelle va émettre des segments qui seront perdus en raison de tentatives infructueuses de transmission, même si l'espacement entre ces tentatives évolue de façon exponentielle.

g) Calcul du débit théorique de données

Pour l'initialisation d'une séquence de transmission de segments, le protocole TCP envoie des données à travers les phases successives de démarrage lent et d'évitement de congestion. Ces phases permettent au protocole de déterminer les paramètres du réseau tel que le débit maximal pouvant être injecté sans congestionner le réseau. L'influence de la période d'indisponibilité du canal se traduit par un retour à la phase de démarrage lent du protocole TCP. Ainsi, l'interruption d'envoi de segments occasionnée par l'indisponibilité du canal engendre une chute dans l'allure de la courbe de débit. A la reprise du trafic par disponibilité du canal, l'allure exponentielle de la courbe du démarrage lent est conservée du fait du recours aux phases successives de "slow start" et de "congestion avoidance".

Lorsque la transmission devient possible entre l'instant $Ts + Tr * X_i$ et l'instant $Ts + Tr * X_{i+1}$, la rafale du "slow start" qui suit la fin de la politique persistante de retransmission commence plus tôt en comparaison avec l'instant de démarrage du trafic de la politique traditionnelle. Dans les deux cas et sous des hypothèses identiques, la durée de la rafale étant la même, puisque la durée de l'interruption a la même influence sur les deux politiques de retransmission, la latence de la politique traditionnelle peut être un intervalle acquis lorsque

celui–ci est suffisamment grand, pour permettre à la politique persistante de faire la différence en injectant des segments supplémentaires. Nous rappelons que, la latence de la politique traditionnelle provient du fait que le protocole TCP attend la prochaine expiration du temporisateur pour effectuer un nouvel envoi sous l'influence d'un état variable du canal sans fil.

Il est donc possible de déterminer le débit théorique supplémentaire minimal (DTM) injecté par la politique persistante en considérant qu'un nouveau segment est injecté dans le réseau au plus tard à chaque période *Tr* initiale de retransmission :

$$DTM = (X_{i+1} - X_i) - a \qquad (14)$$

Ainsi, plus l'intervalle entre les multiplicateurs X_{i+1} et X_i est grand, plus le DTM est grand, de même lorsque le délai *a* est faible.
Nous appelons ce débit, débit théorique minimal du fait qu'il est possible de recevoir un acquittement sans aller au bout de l'intervalle de temps donné par *Tr*, et donc d'injecter un nouveau segment dans le réseau. Les bouts de temps restants peuvent s'accumuler à l'avantage du débit des données de la politique persistante.

Nous avons choisi de comparer les deux politiques de retransmission en conservant le recours à la phase de "slow start" du protocole après un blocage temporaire lié à l'indisponibilité du canal sans fil suivi d'un changement favorable de l'état du canal qui occasionne le démarrage du trafic. Une amélioration de cette implantation de la politique persistante de TCP consistera à éviter le retour à la phase de "slow start", puisque ce non–recours au démarrage lent permet aux algorithmes Reno et NewReno d'améliorer considérablement les performances du protocole.

III.6. Evaluation de performances par Simulation des politiques persistante et traditionnelle de retransmission

L'évaluation de performances à laquelle nous procédons dans cette partie est destinée à comparer par simulation, la politique actuelle de la temporisation de TCP appelée politique traditionnelle, lors de l'envoi de données dans un canal à état variable, avec celle que nous proposons appelée politique persistante. La différence notoire entre les deux politiques de retransmission provient du fait que la politique persistante prend en compte l'état du canal fourni par la couche liaison, au moyen du mécanisme cross–layer. Pour chaque nœud mobile de la simulation, les protocoles IP et DSR sont utilisés comme couche réseau, et le 802.11 aux niveaux MAC et physique.

Les deux politiques de temporisation pour la retransmission des segments sont internes à un nœud donné. Elles sont comparées à travers un modèle d'état du canal lié à un nœud. L'état du canal détermine les conditions de sortie ou pas des paquets du nœud vers l'extérieur du réseau. Il détermine de ce fait la réception ou non de l'acquittement des segments TCP envoyés. L'environnement réseau extérieur à un nœud mobile donné et qui se situe au delà de l'état du canal immédiat du nœud, n'intervient pas dans l'évaluation de performances.

La simulation est faite dans l'environnement ns-2 [FAL03]. Pour sa mise en œuvre, nous avons choisi un scénario simple composé d'un nœud émetteur et d'un nœud récepteur. Le nœud émetteur envoie un trafic CBR avec un temps inter arrivée de paquets durant les pics fixé à 0,04 secondes. Les deux politiques de retransmission sont évaluées avec trois tailles différentes de paquets : 512, 1024 et 2048 octets.

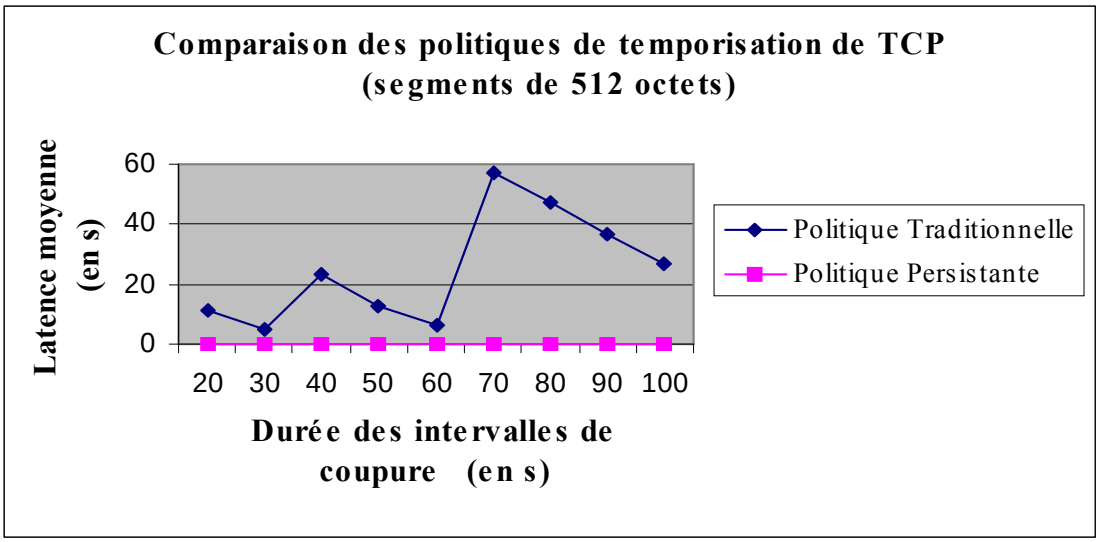
La variation de l'état du canal est matérialisée par un modèle de coupure alternée dans lequel sont définis des intervalles de disponibilité et d'indisponibilité du lien sans fil, de telle sorte qu'un intervalle de disponibilité succède à un intervalle d'indisponibilité. Ces intervalles sont de même taille et commencent à des dates choisies de façon aléatoire entre la 200^{e} seconde et la 300^{e} seconde par une taille de 20 secondes choisie pour dépasser le temps de la première tentative de retransmission de TCP. Le choix aléatoire des dates de début du modèle de coupure alternée s'est fait de façon répétitive dans les 20 simulations exécutées pour chacune des durées de disponibilité/indisponibilité, fixées une à une. Les résultats présentés sont une moyenne de ces 20 simulations de chaque durée fixée de l'intervalle de coupure alternée.

Nous avons fait varier les durées de disponibilité/indisponibilité du canal entre 20 et 100 secondes.

Le scénario intègre une première phase destinée à faire ressortir l'évolution de la période d'attente avant retransmission de la politique traditionnelle de TCP. Le nœud émetteur est placé hors de portée du nœud de destination au lancement de la simulation. Le démarrage du trafic se fait à ce même moment. Le rapprochement des deux nœuds commence à partir de la 100^{e} seconde. Les nœuds se stabilisent lorsqu'ils sont à la portée l'un de l'autre. Le calcul du taux de tentatives infructueuses ne fait pas abstraction de cette portion du temps, il porte sur toute la durée de la simulation.

III.6.1. Latence d'envoi des politiques de temporisation de TCP : résultats de la simulation.

Dans les parties qui suivent, nous présentons les résultats des simulations avec les paquets de 512 octets. Ces résultats ne sont pas éloignés de ceux des autres tailles de paquet. Les courbes de la latence ci–dessous concernent à la fois les paquets de 512 et de 1024 octets. Ils permettent d'illustrer l'intérêt de la temporisation persistante en terme de latence qui a été mise en évidence dans les modèles mathématiques, mais ne permettent pas de procéder à une quantification de cet apport.



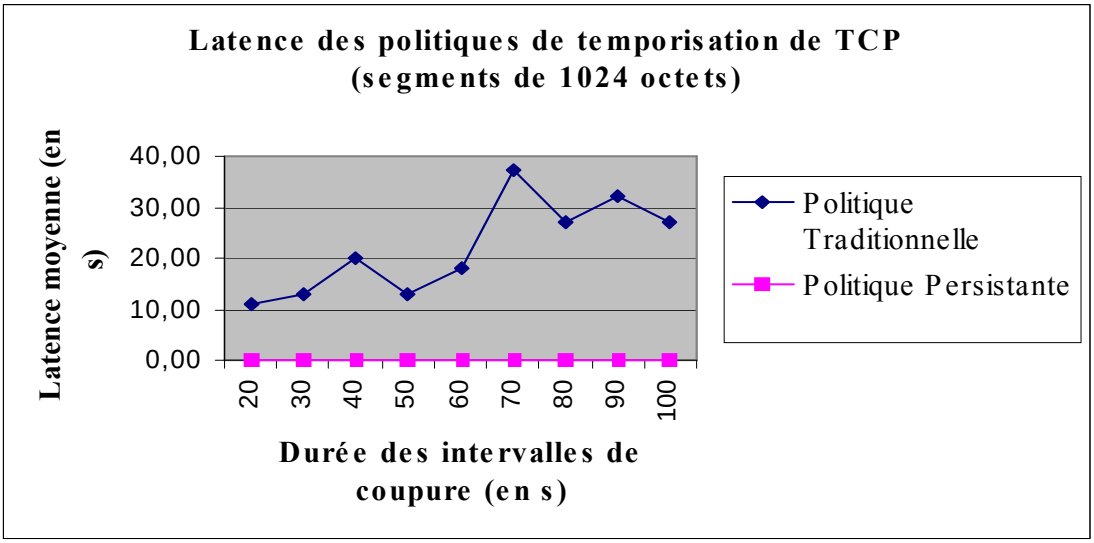


Figure III.4. latence moyenne des politiques de temporisation en fonction des durées d'indisponibilité du canal.

La latence d'envoi est définie comme étant le temps qui s'écoule entre la disponibilité du canal et l'envoi effectif des données par TCP. Les résultats présentés dans la figure III.4 ci–dessus traduisent la latence moyenne observée. L'évolution de la durée de ces intervalles est donné en abscisse.

Les courbent font ressortir un net avantage en faveur de la politique persistante qui envoie le trafic de données dès que le canal le permet, tandis que la politique traditionnelle est handicapée par l'attente de l'expiration du temporisateur. La série illustrée par le modèle de coupure alternée évolue en donnant des bornes ou instants de disponibilité et d'indisponibilité, de façon indépendante des bornes données par les tentatives d'envoi du paquet de la politique traditionnelle. Les résultats de simulation permettent d'observer les déductions intuitives et celles du modèle mathématique qui font que plus la borne de relance est proche de l'instant de disponibilité du canal, plus la latence est faible.

La politique persistante a une latence moyenne de l'ordre d'une fraction de seconde. La moyenne appliquée aux intervalles d'indisponibilité démontre la non linéarité de l'évolution temporelle de la latence de la politique traditionnelle. Les facteurs d'étirement qui sont mis en évidence dans le modèle mathématique n'evoluent pas de façon linéaire. Ceci s'explique par le fait que même sur de très grands intervalles d'indisponibilité du lien, l'indépendance de l'évolution des bornes de disponibilité/indisponibilité par rapport aux bornes de relance de la politique traditionnelle fait que, dans l'évolution du trafic, sur l'un (ou plus) des intervalles de disponibilité, la borne de la tentative d'envoi de la politique traditionnelle soit proche de la borne de disponibilité, ce qui a pour effet de reduire la latence.

III.6.2. Taux de pertes et de tentatives infructueuses (TTI) des politiques de temporisation de TCP : résultats de la simulation.

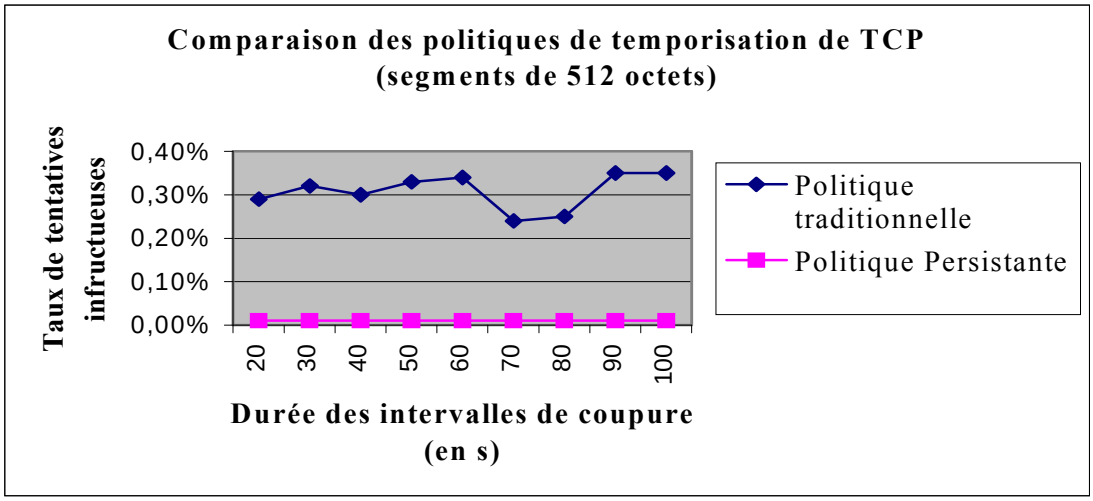


Figure III.5. Taux de perte et de tentatives infructueuses (TTI) des politiques de temporisation en fonction des durées d'indisponibilité du canal.

Le taux de pertes et de tentatives infructueuses est défini comme étant le pourcentage de segments envoyés par la couche transport et perdus par rapport au nombre total de segment envoyés et reçus par le récepteur, y compris les pertes liées au canal. Les courbes de la figure III.5 font ressortir l'avantage intuitif attendu en faveur de la politique persistante et mis en évidence par le modèle mathématique (10). Cet avantage est lié au fait que pour la politique persistante, seul le premier paquet est perdu dans le modèle de coupure alternée, tandis–que la politique traditionnelle envoie un paquet à chaque expiration de son temporisateur. C'est pourquoi le taux *TTI* augmente en fonction de la durée des intervalles d'indisponibilité. Le niveau de la courbe de la politique persistante prend en compte tous les paquets envoyés dans le

scénario par la couche transport, y compris ceux qui sont envoyés et perdus lors du rapprochement des nœuds, puisqu'à ce niveau la politique persistante n'est pas activée. L'exploitation des informations cross–layer qui peuvent être produites par le protocole de routage (DSR ici utilisé) et qui indiquent que la destination n'est pas joignable n'a pas été implantée dans le scénario pour permettre à TCP d'adopter un comportement persistant. Cette partie du scénario a seulement pour avantage de faire ressortir les traces de la politique traditionnelle de retransmission de TCP et n'a pas d'influence sur la comparaison des résultats obtenus.

III.6.3. Consommation d'énergie des politiques de temporisation de TCP : résultats de la simulation.

La politique de contrôle des retransmissions dans le cas du canal sans fil est également sujette aux contraintes de conservation d'énergie. Par exemple, des protocoles de routage orientés prise en compte d'énergie voient le jour tels que le protocole CONSET. La philosophie de base de ce protocole est de diffuser des messages au minimum de puissance requis pour maintenir la connectivité du réseau, sur des longs chemins de bout en bout avec des distances moindres par saut [BHU04]. Ce principe d'utilisation de la puissance minimum est largement pris en compte dans la politique de temporisation persistante.

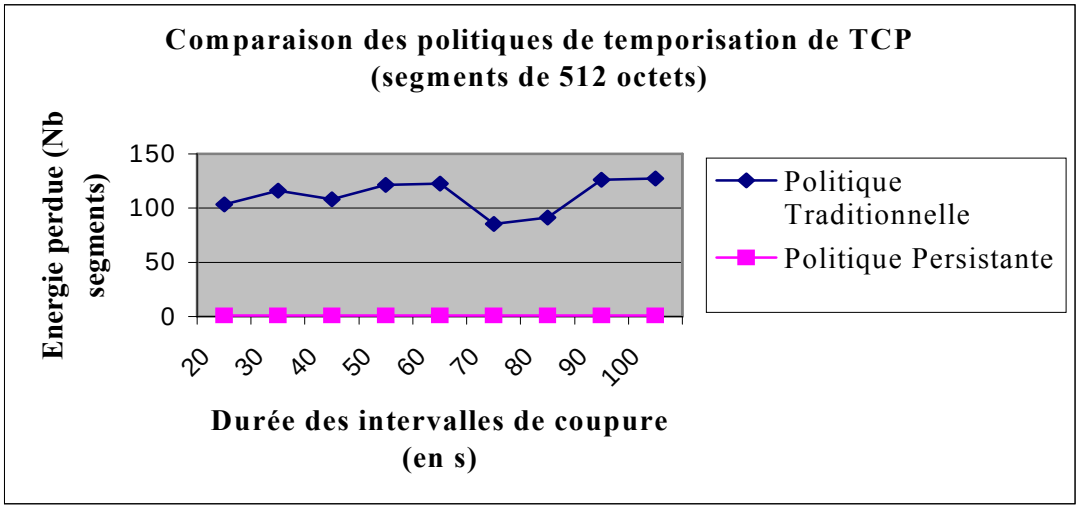


Figure III.6. Consommation d'énergie des politiques de temporisation en fonction des durées d'indisponibilité du canal.

La consommation d'énergie est exprimée ici selon le modèle de Gallager qui fait ressortir le caractère fini de l'energie d'un nœud mobile et qui se traduit par un nombre fini de bits que le noeud peut utiliser avant d'épuiser son énergie. Les courbes de la figure III.7 font ressortir le fait que, la politique traditionnelle qui injecte des segments à chaque expiration de son temporisateur d'envoi, consomme plus d'énergie en nombre de paquets émis et perdus que la politique traditionnelle. Il peut être observé sur les courbes que plus la taille de l'intervalle d'indisponibilité augmente, plus la consommation d'énergie en terme de nombre de segments perdus augmente, comme l'exprime le modèle mathématique en (11) et (12). L'évitement des retransmissions intégré dans la politique persistante a l'avantage de favoriser la diminution de la consommation interne d'énergie occasionnée par l'envoi de segments aux couches inférieures d'un même nœud mobile.

III.7. Conclusion

Dans ce chapitre, nous avons proposé une politique de temporisation destinée à poursuivre l'adaptation du protocole TCP à l'environnement sans fil. La politique persistante ici proposée utilise principalement les informations d'état du canal fournies par la couche liaison à travers le sous–système environnement du modèle conceptuel cross–layer. Etant donné que la politique de temporisation actuelle de TCP, lorsque l'état du canal est mauvais, se traduit par l'envoi de segments TCP à chaque expiration du temporisateur d'attente et surtout se traduit par l'attente de l'expiration de ce temporisateur avant qu'un paquet soit envoyé dans le réseau sans fil même si l'état du canal devient favorable, la politique persistante qui consiste à observer le prochain changement favorable de l'état du canal pour envoyer le segment, dispose d'un avantage démontré dans les modèles mathématiques et traduit par les résultats des simulations en terme de latence, de débit et de taux de tentatives infructueuses coûteuses en terme de consommation d'énergie.

Une autre approche comparative de la politique persistante de temporisation de TCP se fait en intégrant les paramètres d'amélioration de débit dont l'évitement du recours au démarrage lent lors de la reprise du trafic après un blocage temporaire des envois dû à un état défavorable du canal. Ce principe permet d'améliorer les gains obtenus dans les résultats précédents.

La politique persistante de temporisation appliquée au protocole TCP a démontré les gains de performance pouvant être obtenus. Nous procédons dans le chapitre qui suit à l'extension de la même politique de temporisation à l'autre protocole de la couche transport proposé pour les réseaux sans fil, à savoir le protocole SCTP. Ce protocole a la particularité d'émettre des messages réguliers pour maintenir la connectivité à travers le réseau. Il est orienté flux. Les résultats de l'évaluation de performance attendus provenant de l'étude comparée des deux politiques de retransmission, doivent permettre de conforter davantage ceux qui sont obtenus dans ce chapitre, au vu de la différence notoire de comportement qui existe entre les deux protocoles fiables de la couche transport.

Chapitre IV. Extension de la temporisation persistante au protocole SCTP.

IV.1. Introduction

La politique persistante de temporisation appliquée au protocole TCP dans un environnement à canal variable, a produit des résultats de simulation positifs en terme de gain de performance. L'idée relative à la consolidation des acquis de cette politique est d'étendre son application au protocole SCTP [RFC2960] qui est une autre variante de service fiable de la couche transport. Le protocole SCTP comporte un mécanisme de gestion d'association auquel est associé le mécanisme de détection de perte de chemin dans le réseau et la réparation de cette perte grâce à des messages périodiques.

La proposition de la temporisation persistante émane de l'exploitation de l'état variable du canal et repose sur les modèles conceptuels cross–layer produits par application de la méthode RCL au chapitre II. Pour consolider cette politique persistante, nous étendons son application au protocole SCTP, avec au préalable l'utilisation d'une modélisation. L'importance de la méthode de conception pour consolider les acquis de l'architecture est illustrée à travers cette modélisation. L'extension envisagée de la temporisation persistante au protocole SCTP offre un cadre de modification des modèles conceptuels cross–layer produits précédemment lorsque la méthode RCL a été appliquée. Après la production de modèles conceptuels cross–layer, le remplacement du protocole TCP de la couche transport par SCTP, offre une opportunité de modification et de mise à jour de ces modèles, par la prise en compte du nouveau protocole et de nouvelles interactions. Cette extension fait ressortir un autre aspect de cette même méthode.

L'intérêt de la consolidation de la politique de temporisation persistante en l'appliquant au protocole SCTP réside dans la différence de comportement qui existe entre ces deux protocoles de la couche transport. L'un des points communs est qu'ils font tous appel au mécanisme de back–off exponentiel pour la retransmission des messages. Le protocole TCP a été conçu puis optimisé pour l'environnement filaire. A partir des modèles cross–layer, nous avons proposé de le modifier pour prendre en compte l'environnement sans fil. A la différence de TCP, le protocole SCTP a été conçu dès le départ pour servir les mêmes objectifs que le protocole TCP dans l'environnement sans fil avec des variantes supplémentaires comme la gestion des associations et des pertes de chemin.

Après la présentation des variantes de protocoles ordonnés et/ou fiables de la couche transport, nous consacrons une partie de ce chapitre à la présentation du fonctionnement du protocole SCTP, puis à l'application de la méthode RCL à la pile de protocoles modifiée qui remplace TCP par SCTP. La politique persistante de retransmission appliquée au protocole SCTP est par la suite présentée avant l'évaluation de performances par simulation. Une étude comparative des gains obtenus entre les deux protocoles fiables de la couche transport est présentée.

IV.2. Le protocole SCTP

IV.2.1. Positionnement du protocole SCTP

Tout comme le protocole TCP, SCTP assure une transmission fiable et ordonnée de données. UDP [RFC0768] n'assure ni ordre ni fiabilité. Cette notion d'ordre et de fiabilité permet de distribuer la famille des protocoles de la couche transport dans un repère à deux dimensions dont les deux axes représentent les critères d'ordre et de fiabilité, comme l'indique la figure IV.1 ci–dessous. Cette classification [DIA03] a pour objectif de faciliter la compréhension de leur fonctionnement et surtout de matérialiser la différence qui existe entre TCP et SCTP.

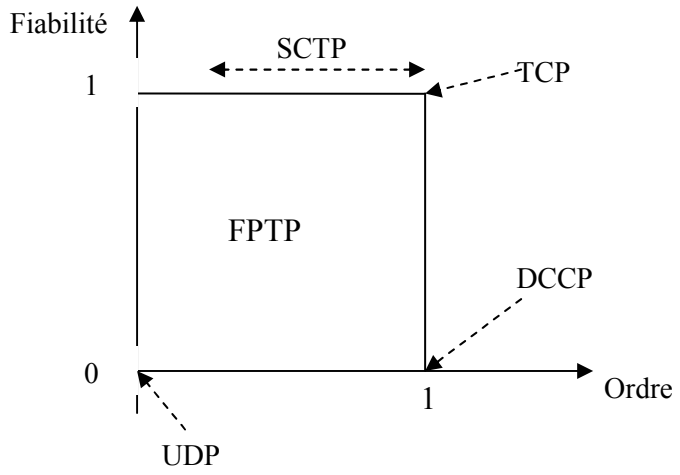


Figure IV.1. Positionnement de la famille des protocoles de la couche transport

Légende :

TCP : Transmission Control Protocol
SCTP : Stream Control Transmission Protocol
FPTP : Fully Programmable Transport Protocol
UDP : User Data gram Protocol
DCCP : Data gram Congestion Control Protocol

Le protocole SCTP utilise la notion de flux et d'association. Un flux est une séquence de messages qui doivent être transmis dans l'ordre. Une association est un concept plus large qu'une connexion TCP, en ce sens qu'il s'agit d'un regroupement de flux dans lesquels chaque point terminal fournit une liste d'adresses de transport (@IP + port).

Le protocole SCTP est un protocole fiable orienté messages fonctionnant au dessus des protocoles sans connexion de la couche réseau tel que le protocole IP. Il offre le service de transfert avec acquittement et sans duplication, le service de fragmentation des données selon la contrainte imposée par la MTU du chemin emprunté. Il offre également un service de livraison ordonnée des messages des utilisateurs à travers plusieurs flux avec une option de délivrance par ordre d'arrivée des messages individuels. Le blocage d'un flux n'a pas d'impact sur la délivrance des autres. SCTP offre également une tolérance aux erreurs provenant de la couche réseau à travers une multi–domiciliation dans différents points terminaux d'une même association. Le protocole SCTP dispose d'un mécanisme d'évitement de congestion, de résistance à la saturation (flooding) et d'attaques déguisées.

IV.2.2. Fonctionnement du protocole SCTP

Deux principes moteurs facilitent la compréhension de l'activité du protocole SCTP. D'une part, cette activité se caractérise par la réponse à des événements qui surviennent tels que les appels utilisateur, l'arrivée de segments et la gestion des temporisateurs. D'autre part, la réaction du protocole par rapport à chaque événement dépend de l'état dans lequel il se trouve. Ces deux principes font de l'implantation de SCTP un ensemble de modules événementiels avec changement d'état.

Dans les points ci–dessous, nous faisons ressortir ce comportement dynamique du protocole SCTP à travers quelques–uns de ses mécanismes. SCTP utilise le mécanisme d'établissement d'une association pour transférer ses données et le mécanisme d'arrêt de cette association à la fin de la session. Il utilise également les mécanismes de gestion des erreurs, d'envoi de données et celui de gestion de la défaillance de chemin. Tout comme le protocole TCP, le protocole SCTP a recours à la phase de "Slow Start" et de "Congestion Avoidance". Il utilise le mécanisme de temporisation pour gérer divers événements tels que la fin d'une association ou la détection d'erreurs de transmission.

IV.2.2.1. Etablissement et arrêt d'une association SCTP

Avant de procéder au transfert de données, l'émetteur et le récepteur SCTP exécutent des séquences d'initialisation pour mettre en place une association. Après l'établissement de l'association, des flux unidirectionnels sont ouverts pour le transfert des données entre les différents points terminaux. Lorsque les étapes ont été franchies avec succès, un objet appelé TCB (Transmission Control Block) est créé au niveau de chaque nœud pour matérialiser l'association à travers laquelle les données pourront être échangées.

L'arrêt d'une association SCTP peut se faire à travers l'envoi d'un message spécial (ABORT) sans option d'accompagnement. Ce message implique une suppression de toute donnée en cours de transmission. Un nœud donné qui reçoit un message contenant cette option d'arrêt de l'association ne doit pas y répondre sans soumettre ce message reçu à une phase de vérification d'étiquette. Le récepteur supprime l'association de ses enregistrements et reporte l'arrêt à la couche supérieure lorsque la vérification a été réalisée avec succès.

IV.2.2.2. Politique traditionnelle de temporisation pour la fin d'une association SCTP

Une association peut être terminée lorsque la couche application envoie le message de fin d'association appelé SHUTDOWN au protocole SCTP. Sur réception de cette primitive, un nœud émetteur entre dans l'état appelé SHUTDOWN-PENDING et y demeure jusqu'à la transmission avec acquittement des données en attente d'envoi. Après réception des acquittements de toutes les données envoyées, le nœud envoie un message SHUTDOWN à son vis à vis et démarre le temporisateur T2-shutdown, puis passe dans l'état SHUTDOWN-SENT. A l'expiration du temporisateur du nœud émetteur, un autre SHUTDOWN est envoyé, ainsi de suite jusqu'à expiration du nombre de tentatives.

A la fin de la réception des données, le nœud récepteur qui reçoit un SHUTDOWN, envoie un SHUTDOWN-ACK, démarre son temporisateur T2-shutdown et passe à l'état SHUTDOWN-ACK-SENT. A l'expiration du temporisateur du récepteur, un autre SHUTDOWN-ACK est envoyé, et ainsi de suite jusqu'à épuisement du nombre de tentatives. Si le nombre de tentatives expire, le récepteur supprime le TCB, notifie à la couche supérieur l'inaccessibilité du vis à vis et fait entrer l'association dans l'état CLOSED.

La réception du SHUTDOWN-ACK est traitée par le nœud émetteur par l'arrêt de son temporisateur T2-shutdown, par le renvoi d'un message SHUTDOWN-COMPLETE et par la suppression de l'association en cours. Lorsque le nœud récepteur reçoit le message SHUTDOWN-COMPLETE, il vérifie s'il est dans l'état SHUTDOWN-ACK-SENT et dans ce cas, il arrête le temporisateur T2-shutdown, envoie un SHUTDOWN-COMPLETE et supprime l'association.

La gestion des temporisateurs T2-shutdown des nœuds émetteur et récepteur se fait par application de la politique traditionnelle de retransmission dans laquelle le temporisateur d'attente avant retransmission prend des valeurs croissantes multiples des valeurs précédentes pour limiter le nombre de tentatives infructueuses. La gestion de la fin d'une association présentée dans les paragraphes précédents est de ce fait sujette à l'optimisation proposée pour remplacer la temporisation traditionnelle par la temporisation persistante dans le cas où le prolongement du mauvais état du canal engendre une impossibilité d'envoi. Dans ce cas, l'impossibilité d'envoi se traduit par l'absence de paquet SHUTDOWN-ACK en réponse à un paquet SHUTDOWN.

IV.2.2.3. Envoi des données

Les deux protocoles TCP et SCTP ont un fonctionnement différent. Ils comportent cependant de multiples points de similitude, d'où l'intérêt d'effectuer une évaluation comparative des performances de la temporisation persistante au niveau des deux protocoles.

L'échange de grand volume de données en utilisant le protocole SCTP est similaire à la procédure d'acquittement sélectif du protocole TCP. La réception d'un grand volume de données est acquittée par l'envoi d'un segment SACK qui indique non seulement le TSN (numéro de séquence de transmission) cumulé reçu, mais aussi les TSN non cumulés comportant des intervalles (gaps). Comparativement aux procédures du protocole TCP, les SACK sont envoyés via la méthode d'"ACK différé", normalement avec un SACK par segment reçu, mais avec une limite supplémentaire sur les délais entre les SACK et une augmentation pour les segments contenant des gaps.

Par ailleurs, le contrôle de flux et le contrôle de congestion sont les mêmes que ceux de l'algorithme du protocole TCP. La fenêtre d'avertissement indique l'occupation du tampon du récepteur tandis que la fenêtre de congestion par chemin est maintenue pour gérer les paquets en vol. Le "Slow Start", l'évitement de congestion, le Fast Recovery et le Fast Retransmit sont intégrés dans les procédures comme décrit dans [RFC2581]. Une modification y est introduite et se rapporte au fait que les points terminaux doivent gérer la conversion entre les octets envoyés et reçus et les TSN envoyés et reçus, étant donné que les TSN sont établis pour de grands volumes de données plutôt que par octet. L'application peut spécifier la durée de vie des données à envoyer de telle sorte que si cette durée de vie expire et que les données n'ont pas encore été transmises, elles peuvent être détruites (exemple : messages de signalisation sensibles au delai). Si les données ont été transmises, la livraison doit se poursuivre pour éviter un trou dans la séquence de TSN.

IV.2.2.4. Gestion des erreurs du protocole SCTP

Le protocole SCTP utilise le mécanisme de retransmission pour gérer les erreurs de transmission. Les retransmissions des données sont déclenchées soit par l'expiration du temporisateur de retransmission soit par la réception d'un SACK indiquant que les données n'ont pas été reçues. Pour réduire la congestion potentielle, la fréquence des retransmissions de données est limitée. Le temporisateur de retransmission RTO (Retransmission Time Out) est ajusté sur la base de l'estimation du "round trip delay" et décrit un "back–off" exponentiel lorsque la perte des messages augmente. Le "back–off" exponentiel signifie que le temporisateur prend une valeur multiple de sa valeur précédente pour chaque tentative infructueuse d'envoi, tout comme le fait le protocole TCP.

Dans une association active avec une transmission équitable et constante de données, les SACK génèrent plus de retransmissions que l'expiration du temporisateur. Pour réduire la possibilité d'avoir des retransmissions non nécessaires, la règle de quatre SACK est utilisée, de telle sorte que la retransmission n'est déclenchée qu'à la réception du quatrième SACK indiquant que les données sont perdues. Ceci est censé éviter la retransmission de messages dans de nombreux cas tels que le déséquencement.

IV.2.2.5. Défaillance du chemin

Le protocole SCTP utilise au niveau transport un mécanisme de détection de perte de chemin qui est un événement caractéristique des réseaux mobiles sans fil. Pour détecter qu'un chemin du réseau est indisponible, le protocole établit un décompte du nombre de retransmissions sans réception d'acquittement pour une adresse de destination donnée. Lorsque ce décompte dépasse un maximum prédéfini, l'adresse est déclarée inactive et une notification est faite à l'application. Dans un tel cas et pour un nœud multi–domicilié, le protocole utilisera par la suite l'adresse alternative si elle est disponible pour envoyer ses données.

La détection de la perte de chemin ne se fait pas seulement au moyen du décompte du nombre de retransmissions. Des messages périodiques appelés "Heartbeat" sont envoyés à intervalle régulier pour toutes les destinations disponibles (par exemple, les différentes adresses d'un même nœud). Le protocole SCTP maintient un compteur qui indique le nombre de "Heartbeat" envoyés à une destination indisponible sans réception d'acquittement. Lorsque ce compteur dépasse un maximum prédéfini, l'adresse de destination en question est déclarée inactive. Les "Heartbeat" continuent d'être envoyés aux destinations inactives jusqu'à ce qu'un ACK soit reçu. Dans ce cas, l'adresse est désignée comme étant à nouveau active. Le taux d'envoi des "Heartbeat" est associé à l'estimation du RTO auquel est ajouté un paramètre de délai qui permet au trafic "Heartbeat" de s'adapter aux besoins des applications.

Pour déduire l'indisponibilité d'un point terminal, le protocole SCTP maintient un compteur pour chaque adresse de destination. Ce compteur indique le nombre de retransmissions ou de "Heartbeat" envoyés sans réception d'acquittement. Lorsqu'un compteur dépasse un maximum prédéfini, le point terminal est déclaré inaccessible, de ce fait, l'association SCTP est alors fermée.

De part l'étude des mécanismes cross–layer, le SCTP met en œuvre au niveau transport des mécanismes de détection des voisins qui deviennent redondants lorsque des protocoles de routage qui émettent des paquets réguliers de signalisation sont utilisés au niveau réseau.

IV.3. Modèles conceptuels cross–layer du protocole SCTP : Application de la méthode RCL

L'application de la méthode RCL sur la nouvelle pile de protocoles consolide l'utilisation de la méthode et comporte des intérêts multiples dont le respect des acquis de l'architecture. Un autre intérêt de l'utilisation de la méthode RCL est de permettre d'évaluer la complexité de l'adaptation à réaliser pour que la nouvelle pile de protocoles intègre les nouvelles interactions cross–layer qui seront recensées. Cette évaluation est faite à travers les modèles conceptuels produits par application des étapes successives de la méthode. Un autre intérêt de cette application réside dans la mesure des modifications que les points de différence de fonctionnement entre les deux protocoles ont engendrés sur les modèles conceptuels produits. Enfin, les modèles conceptuels cross–layer mis à jour feront ressortir les mécanismes de la pile de protocoles qui apparaissent en double et qui peuvent être remplacés par de nouveaux mécanismes cross–layer unifiés ou simplifiés.

IV.3.1. Choix de la pile des protocoles

A cette étape, nous modifions la pile de protocoles sur laquelle la méthode RCL a été précédemment appliquée. Le protocole TCP est ainsi remplacé par SCTP au niveau transport. Le protocole DSR et le protocole IP sont maintenus au niveau réseau, de même que le protocole IEEE 802.11 pour les couches basses.

Le choix de la pile de protocole sur laquelle la méthode RCL est appliquée est suivi par l'étape de recensement des différentes interactions AACL utilisées ou produites par chaque protocole.

IV.3.2. Recensement des AACLs

Le recensement à effectuer respecte la classification prédéfini des AACL. Elles seront regroupées selon leur catégorie, à savoir les AACL de services "Activables", les AACL de "Mise à disposition" et les AACL de "Notification". Une simple règle de restriction est appliquée pour éviter une redondance improductive. Comme les modifications de la pile de protocoles ne concernent que la couche transport, seules les AACL utilisées ou générées par les protocoles de transport seront considérées à nouveau. La règle d'ajout de nouvelles AACL découle du caractère exhaustif du recensement. Ce recensement est rendu nécessaire par les modifications liées au remplacement du protocole TCP par le protocole SCTP dans la pile de protocoles. La règle de conservation des AACL est la troisième règle considérée à cette étape. Une AACL existante qui n'est pas utilisée par le protocole remplacé mais qui l'est par le nouveau protocole sera conservée.

Le protocole SCTP utilise certaines AACL de "Notification" dont les données étaient précédemment consommées par TCP. Il s'agit notamment de la "Notification" d'acquittement de la couche liaison, la "Notification" d'évitement de retransmission, la "Notification" de la gigue d'envoi des paquets, la "Notification" de la baisse significative du niveau d'énergie, la "Notification" de la gigue d'envoi due à la défaillance d'une route, la "Notification" de la gigue d'envoi due au changement de route. Ces AACL sont conservées avec le même principe d'exploitation que pour TCP. La liste de ces AACL conservées est donnée par les points de similitude de fonctionnement entre les deux protocoles.

La nouvelle AACL ajoutée à la catégorie ci–dessus concerne la "Notification" de la défaillance d'un chemin. Cette AACL est fournie par le mécanisme SCTP d'émission périodique de paquet "heartbeat" grâce auquel le protocole détecte la disponibilité ou non d'une destination dans le réseau. Cette notification se fait uniquement à destination de la couche application du fait qu'il est supposé que les paquets "heartbeat" émis sont acheminés par les primitives du protocole de routage et que l'indisponibilité du nœud sera détectée au niveau du routage et permettra ainsi la mise à jour des tables.

Il existe des AACL de "Mise à disposition" utilisées par TCP qui sont également conservées pour être utilisées par SCTP avec le même principe d'exploitation. Ce sont les AACL de "Mise à disposition" du taux de perte de paquet, de "Mise à disposition" du SNR, de "Mise à disposition" du taux d'erreur bit BER, de "Mise à disposition" de la puissance du signal reçu RSS et de "Mise à disposition" du niveau d'énergie.

L'analyse de la portée de chaque interaction nous permet de conserver deux AACL de services activables utilisées par TCP. Les AACL de Service FEC "Activable" et de Service ARQ "Activable" ont un impact sur le fonctionnement du protocole SCTP.

IV.3.3. Tableau des interactions des protocoles par AACL

La création du tableau des interactions des protocoles respecte la logique de la modification apportée à la pile de protocoles. Les AACL qui apparaissent dans le tableau sont celles qui ont été retenues pour l'impact qu'elles ont sur le protocole SCTP.

Action Atomique Cross–layer (AACL)	**Protocoles**					
	Application	**SCTP**	**DSR**	**IP**	**Liaison 802.11**	**Physique 802.11**
"Notification" de l'indisponibilité d'un nœud	D	S				
"Notification" de la gigue d'envoi des paquets		D			S	
"Notification" d'évitement de retransmission		D	D		S	
"Notification" d'acquittement		D3	D2, S3		D1, S2	S1
"Notification" de la baisse significative du niveau d'énergie	D	D	D	D	D	D
"Notification" de la gigue d'envoi due à la défaillance d'une route		D	S			
"Notification" de la gigue d'envoi due au changement de route		D	S			
"Mise à disposition" du taux de perte de paquet	U	U			S	
"Mise à disposition" du SNR	U	U			U	S
"Mise à disposition" du RSS	U	U			U	S
"Mise à disposition" du taux d'erreur bit BER	U	U			U	S
"Mise à disposition" du niveau d'énergie	U	U	U	U	U	U
Service FEC "Activable"		U			S	
Service ARQ "Activable"		U			S	

Table IV.1. Tableau des interactions des protocoles par AACL (TCP est remplacé par SCTP)

IV.3.4. Tableau d'interaction des fonctions SCTP par AACL

La philosophie fonctionnelle de base des deux protocoles TCP et SCTP est très voisine, mais l'une de leurs différences principales réside dans les séquences de mise en œuvre de cette philosophie. Le protocole SCTP assure un transfert fiable de données à travers une association entre des nœuds du réseau sans fil. Tout comme le protocole TCP, SCTP assure une fonction de contrôle de données transférées qui permet d'établir si les données transférées sont endommagées, perdues ou dupliquées ou encore déséquencées. Il assure également la fonction de correction d'erreur à travers le mécanisme de retransmission, la fonction de contrôle de flux grâce à l'usage de la fenêtre de transmission, la fonction de contrôle de congestion grâce au mécanisme d'évitement de congestion et aussi la fonction de gestion de priorité par le mécanisme de transmission prioritaire des données. Une autre fonction du protocole SCTP inexistante au niveau de TCP concerne la gestion de la défaillance de chemin que l'on retrouve traditionnellement au niveau des protocoles de routage.

Le tableau suivant est produit sur la base de cette subdivision du protocole SCTP en différentes fonctions. Il représente le modèle conceptuel d'interactions entre les fonctions du protocole SCTP et les AACL recensées. En référence à la méthode RCL, ce tableau présente les AACL utilisées par les différentes fonctions du protocole SCTP ainsi que les autres protocoles qui interviennent. Le tableau d'interaction des fonctions permet par exemple de fixer la fonction de gestion de la défaillance de chemin comme source effective de l'AACL de "Notification" de l'indisponibilité d'un nœud. Les autres AACL restent conformes à l'esprit de la répartition fonctionnelle faite au niveau du protocole TCP. C'est l'exemple de l'AACL de notification de la gigue d'envoi des paquets qui indique que la fonction de contrôle de données transférées est la destination effective de cette AACL initiée par le protocole 802.11 au niveau liaison. L'étape six de la méthode RCL décrit la modification envisagée pour la prise en compte de la notification de la gigue d'envoi des paquets par la fonction de SCTP qui réalise le contrôle de données transférées.

Actions Atomiques Cross–layer	Fonctions SCTP						Autres Protocoles				
	Contrôle données transféré.	**Correct. d'erreur**	**Contrôl de congest.**	**Gestion de priorité**	**Gest° de flux**	**G. déf. chemin**	**Applica tion**	**DSR**	**IP**	**Liaison 802.11**	**Physique 802.11**
"Notification" de l'indisponibilité d'un nœud						S	D				
"Notification" de la gigue d'envoi des paquets	D									S	
"Notification" d'évitement de retransmission	D							D		S	
"Notification" d'acquittement	D3							D2, S3		D1, S2	S1
"Notification" explicite de congestion			D						S dist.		
"Notification" de la baisse significative du niveau d'énergie	D							D	D	D	S
"Notification" de la gigue d'envoi due à la défaillance d'une route	D							S			
"Notification" de la gigue d'envoi due au changement de route	D							S			
"Mise à disposition" du taux de perte de paquet	U									S	
"Mise à disposition" du SNR	U									U	S
"Mise à disposition" du RSS	U									U	S
"Mise à disposition" du taux d'erreur bit BER	U									U	S
"Mise à disposition" du niveau d'énergie	U							U	U	U	S
Service FEC "Activable"	U									S	
Service ARQ "Activable"		U								S	

Table IV.2. Tableau d'interaction des fonctions SCTP par AACL

IV.3.5. Déduction de(s) modèle(s) d'interaction des AACL

L'étape 5 de la méthode RCL permet de produire les modèles d'interaction des AACL par catégorie. Ces modèles d'interaction font ressortir l'utilisation des sous–systèmes engendrés par les interactions ainsi que les mécanismes cross–layer internes aux couches réseaux. Ils permettent de juger de la complexité des échanges lors de la mise en œuvre des modèles d'interaction. La déduction du degré de complexité des modèles d'interaction reste une règle intuitive.

Les modifications apportées à la pile initiale de protocoles utilisant le protocole TCP sont mineures. Ces modifications qui ont trait au remplacement de TCP par SCTP se sont traduites par :

- l'ajout d'une fonction aux modèles donnés par TCP concernant la gestion de la défaillance de chemin,
- le recensement d'une AACL supplémentaire qui est la "Notification" de l'indisponibilité d'un nœud.

Les modèles d'interaction des AACL engendrés par les deux piles de protocoles sont voisins. C'est pourquoi, nous faisons abstraction de la présentation des modèles d'interaction des AACL engendrés par la pile qui utilise SCTP à la place de TCP. Tout comme les modèles d'interaction de la première pile de protocoles, ceux de la deuxième pile sont également de moindre complexité.

IV.3.6. Tableau descriptif des interactions : cas du protocole SCTP

L'étape 6 de l'application de la méthode RCL permet de déduire le tableau de description des interactions de chaque protocole appartenant à la pile de l'étape 1. Le tableau suivant présente l'exploitation qui est faite de chaque AACL par la fonction de chaque protocole.

Ce tableau présente la description des interactions du protocole SCTP. Les lignes indiquent les propositions d'utilisation de chaque AACL par les fonctions du protocole SCTP ainsi que les modifications du code source de ces fonctions.

AACL	Fonction SCTP	Exploitation au niveau du SCTP
"Notification" de la gigue d'envoi des paquets	Contrôle données transférées	Réinitialiser le compteur d'attente du SACK du segment, Pas de retransmission de ce segment pour la nouvelle durée (ne pas incrémenter le compteur des retransmissions pour la même destination), Pas d'émission de segment "heartbeat" pour la nouvelle durée, Ne pas invoquer le mécanisme de contrôle de congestion.
"Notification" de la gigue d'envoi due à la défaillance d'une route		
"Notification" de la gigue d'envoi due au changement de route		
"Notification" d'évitement de retransmission		Geler les transmissions et retransmissions pour la durée spécifiée dans le message. Réinitiatiser les temporisateurs de retransmission sans incrémenter le compteur.
"Notification" de l'indisponibilité d'un nœud		Selon la valeur du compteur de "heartbeat" ou de retransmission, mettre à jour l'état joignable ou non d'une destination dans le sous–système environnement
"Notification" d'acquittement		Anticiper la transmission de nouvelles données s'il est établi par le DSR que le destinataire est à portée directe.
"Notification" explicite de congestion	Ctrl congestion	Invoquer le mécanisme de contrôle de congestion.
"Notification" de la baisse significative du niveau d'énergie	Contrôle données transférées	Modifier la fréquence de retransmission et/ou le débit de transmission, Désactiver l'usage du mécanisme d'émission de paquets "heartbeat".
"Mise à disposition" du taux de perte de paquet		Ajuster le débit de transmission et la fréquence des retransmissions suivant les valeurs de ces paramètres établies à partir de seuils (indiquent l'état du canal). Alléger l'usage du mécanisme d'émission de paquets "heartbeat".
"Mise à disposition" du SNR		
"Mise à disposition" du taux d'erreur bit BER		
"Mise à disposition" du RSS		Utiliser l'ACK de la couche liaison à titre de SACK du paquet transmis si le seuil de ce paramètre indique que le nœud en vis à vis est à portée directe
"Mise à disposition" du niveau d'énergie		Ajuster le débit de transmission, Modifier la fréquence de retransmission, désactiver l'usage du mécanisme d'émission de paquets "heartbeat" suivant les valeurs de ce paramètre établies à partir de seuils.
Service FEC "Activable"		Désactiver le mécanisme de vérification des données reçues (vérification du checksum).
Service ARQ "Activable"	Correction d'erreurs	Désactiver la fonction de correction d'erreur données s'il est établi par DSR que le destinataire est à portée directe.

Table IV.3. Tableau de description des interactions du protocole SCTP

IV.4. Politique persistante de retransmission

Les modèles conceptuels cross–layer traduisent les acquis de l'application de la méthode RCL. En remplaçant TCP par SCTP, la principale conclusion qui apparaît de ces modèles conceptuels est l'indication de la convergence de réaction aux AACL des deux protocoles. Le protocole SCTP utilise le mécanisme de retransmission à divers niveaux, notamment lors de la transmission de messages de données et lors de la transmission de messages de contrôle. Dans chaque cas, l'expiration du temporisateur de retransmission ou la réception d'un SACK indiquant la non–réception d'un paquet sont les principaux événements déclencheurs des retransmissions. La retransmission déclenchée par la réception d'un SACK est une retransmission sélective de messages et est fondamentalement différente de la retransmission engendrée par l'absence de SACK. Cette dernière retransmission est déclenchée par l'expiration du temporisateur d'envoi. La politique de retransmission qui est utilisée que nous appelons politique traditionnelle est la même que celle du protocole TCP, à savoir le back–off exponentiel. C'est pourquoi, en raison de l'utilisation de ce back–off exponentiel dans la gestion des retransmissions (politique traditionnelle), il devient possible d'étendre la temporisation persistante au mécanisme de gestion d'erreur utilisé par le protocole SCTP avec la même philosophie fonctionnelle que celle du protocole TCP.

Comme pour TCP, le back–off exponentiel décrit par le mécanisme de retransmission de SCTP est comparable au fonctionnement des protocoles non persistants de niveau MAC. Ces protocoles non persistants imposent un délai d'attente aléatoire lorsque le canal est occupé avant la prochaine tentative. A la différence de cette philosophie fonctionnelle, les protocoles persistants continuent d'observer le canal et émettent la trame dès qu'il est libre. L'état du canal sans fil est variable, ce qui gêne la transmission des données quand son état est défavorable. Les modèles cross–layer permettent de rendre l'état du canal accessible à toutes les couches à travers le sous–système environnement. Le prolongement d'un mauvais état du canal engendre immanquablement le déclenchement des retransmissions du fait de l'expiration du temporisateur d'attente d'acquittement.

Par analogie avec le mécanisme de la couche MAC pour lequel l'émission d'une trame ne doit se faire que si le canal est libre, la retransmission d'un segment au niveau transport n'est efficace que si l'état du canal le permet. En observant l'évolution des intervalles de retransmission déclenchées par l'expiration du temporisateur, les prochaines tentatives de ré–émission de segment dans le déroulement du back–off exponentiel sont intimement liées à l'expiration du temporisateur qui prend une valeur croissante multiple de sa précédente à chaque expiration. De ce fait, dans cette politique traditionnelle, lorsque l'état du canal bloque temporairement les transmissions de données, la prochaine tentative qui suit le changement favorable de l'état du canal ne se fera qu'à l'expiration du temporisateur, ce qui ajoute une latence de transmission. Un autre inconvénient de la politique traditionnelle lorsqu'un mauvais état du canal bloque temporairement l'envoi des données provient du nombre croissant de tentatives infructueuses d'émission. Ces tentatives ne sont pas négligeables en terme de consommation d'énergie. La consommation d'énergie détermine la durée de vie du réseau sans fil et revêt de ce fait un aspect primordial au sein de ces réseaux.

Pour optimiser la latence d'envoi, réduire au minimum le nombre de tentatives infructueuses consommatrices d'énergie, il est possible d'appliquer au protocole SCTP le principe de retransmission persistante par lequel, lorsqu'un mauvais état du canal bloque temporairement l'envoi des données, le protocole observe continuellement la variation de l'état du canal et reste ouvert à la notification explicite de changement favorable de cet état, en lieu et place du principe de back–off exponentiel. En outre, contrairement à la couche MAC, le niveau transport appliquant cette politique persistante n'est pas directement soumis à des risques de collision.

IV.5. Evaluation de performances par simulation

IV.5.1. Description des scénarios

Le scénario simulé est le même que précédemment. La simulation est faite dans l'environnement ns–2 avec un nœud émetteur et un nœud récepteur. Le trafic constant du nœud émetteur est soumis à la variation de l'état du canal. Le temps inter–arrivée de messages durant les pics est fixé à 0,1 seconde. Pour chacun des protocoles TCP et SCTP, les deux politiques de retransmission traditionnelle et persistante sont évaluées avec des segments de 1500 octets.

Le modèle d'état du canal est le même. Dans ce modèle, à partir de la 200^{e} seconde de simulation, un intervalle de disponibilité succède à un intervalle d'indisponibilité avec au total 5 intervalles de disponibilité du lien et 5 autres intervalles d'indisponibilité. Ce principe de succession alternée permet de faire correspondre l'indisponibilité du lien sans fil à divers moments de l'évolution du trafic.

Les résultats de la simulation présentés sont ceux des 9 cas de coupure alternée dans lesquels les intervalles de disponibilité et d'indisponibilité du lien sont compris entre 20 et 100 secondes. Le scénario initial dans lequel les intervalles portent sur 20 secondes est choisi pour dépasser le temps de la première tentative de retransmission des deux protocoles TCP et SCTP. L'évolution de 10 secondes que nous avons par la suite appliquée d'un scénario à un autre permet d'éviter des intervalles trop proches. Chaque scénario est évalué 20 fois avec à chaque fois un glissement de 5 secondes de la date de début de la coupure alternée.

L'objectif de la simulation est de comparer les performances des deux protocoles TCP et SCTP, lorsqu'ils sont soumis aux aléas d'un canal à état variable et qu'ils répondent à la variation de l'état de ce canal en adoptant le principe de la temporisation persistante. Les scénarios simulés sont rigoureusement identiques pour permettre de procéder à la comparaison.

IV.5.2. Base de l'interprétation des résultats

Nous allons comparer les performances des deux protocoles TCP et SCTP relativement à l'utilisation de la politique persistante de retransmission en réponse à l'indisponibilité du canal. Les critères de comparaison sont les mêmes que dans les précédentes simulations :

- Nous évaluons la latence de l'envoi effectif des données lorsque le canal devient disponible après une période de suspension conforme aux intervalles de coupure alternée.

- Le critère de débit théorique maximal n'est pas présenté mais est lié à l'intuition par laquelle l'amélioration des temps de latence doit permettre aux protocoles d'injecter un trafic supplémentaire.

- Le critère des émissions infructueuses de messages est lié à la nature des politiques traditionnelles de retransmission qui est un back–off exponentiel. L'apport de la politique persistante est mesuré en terme de nombre de retransmissions de chaque protocole en fonction de la politique retenue.

- La consommation d'énergie est un autre critère utilisé pour comparer les deux politiques de retransmission de TCP et SCTP. Cette consommation est considérée sous l'angle de la définition faite dans les travaux de Gallager. L'auteur défini un nœud mobile disposant d'une source d'énergie finie comme ayant un nombre fini de bits qu'il peut transmettre avant d'épuiser son énergie. L'émission de paquets d'une couche à une autre est une activité tout aussi consommatrice d'énergie. C'est pourquoi ces émissions par les politiques de retransmission tombent sous le coup de la loi de Gallager. Nous procéderons à une comparaison de l'énergie consommée par chaque politique de retransmission.

IV.5.3. Courbes de Latence (TCP et SCTP)

Les courbes de la figure IV.2 ci–dessous permettent de comparer les latences moyennes des politiques traditionnelle et persistante de temporisation des protocoles TCP et SCTP. Pour chaque protocole pris isolement, la courbe de la politique traditionnelle affiche une latence moyenne supérieure à celle de la politique persistante. Ces résultats affichés confirment les observations effectuées précédemment au chapitre III. L'analyse croisée de ces résultats permet d'effectuer une comparaison de l'influence du canal à état variable sur les mécanismes de chaque protocole.

La première étape de l'analyse croisée considère le calcul de la latence de la politique traditionnelle des deux protocoles. Les résultats de la latence en politique traditionnelle reflètent le comportement du protocole SCTP qui envoie régulièrement les paquets de signalisation HeartBeat. Ces envois réguliers permettent au protocole de détecter rapidement le changement d'état du canal sans attendre l'expiration de son temporisateur comme le fait TCP. La rapidité de détection de ce changement est un avantage lié à l'utilisation de l'envoi périodique des paquets HeartBeat. En revanche, le protocole est handicapé par l'établissement de la disponibilité du nœud de destination par l'échange de ces mêmes messages de signalisation avant de procéder au transfert proprement dit des données. Cette phase de certification de la disponibilité du nœud de destination rallonge la latence du protocole SCTP. La base de comparaison est faite à partir de l'envoi effectif des données. Malgré des différences, les courbes de latence en politique traditionnelle donnent des résultats mitigés, en faveur de l'un ou l'autre des deux protocoles, en fonction des moments de disponibilité du canal, de l'expiration du temporisateur de retransmission de TCP et de la fin de l'échange des paquets Heartbeat du protocole SCTP.

La seconde étape de l'analyse croisée se rapporte au calcul de la latence de la politique persistante des deux protocoles. Le mécanisme d'émission régulière de données à chaque expiration du temporisateur est remplacé par un comportement persistant. Les deux protocoles observent le prochain changement favorable de l'état du canal avant de procéder à l'envoi des données. Le mécanisme de gestion de la défaillance du chemin est aussi modifié pour prendre en compte l'état variable du canal. Ce mécanisme est désactivé lorsque le canal est dans un mauvais état et est réactivé sitôt l'état du canal redevenu favorable, en même temps que le

mécanisme d'envoi des données. Par conséquent, pendant la phase de persistance, l'envoi des messages Hearbeat est également suspendu. Le comportement observé est que le protocole fait recours à la certification de la disponibilité du nœud de destination par l'échange des paquets Heartbeat avant d'envoyer les données. C'est pourquoi la latence moyenne en politique persistante est améliorée par TCP du fait qu'il procède à l'envoi des données sitôt que l'état du canal devient favorable tandis que SCTP est handicapé par l'échange des messages HeartBeat avant l'envoi des messages de données.

L'utilisation du Heartbeat sert à tester l'accessibilité d'une destination. Dans notre cas, les deux nœuds sont en visibilité directe. Par conséquent, l'information du "canal bon" suffit pour garantir l'émission d'information. C'est pour cela que dans notre cas de figure, TCP avec la temporisation persistante est meilleur que SCTP qui est ralenti pas les messages Heartbeat. Inversement, dans le cas de la politique traditionnelle, les messages Heartbeat retrouvent tout leur intérêt et conduisent à des meilleurs résultats pour SCTP.

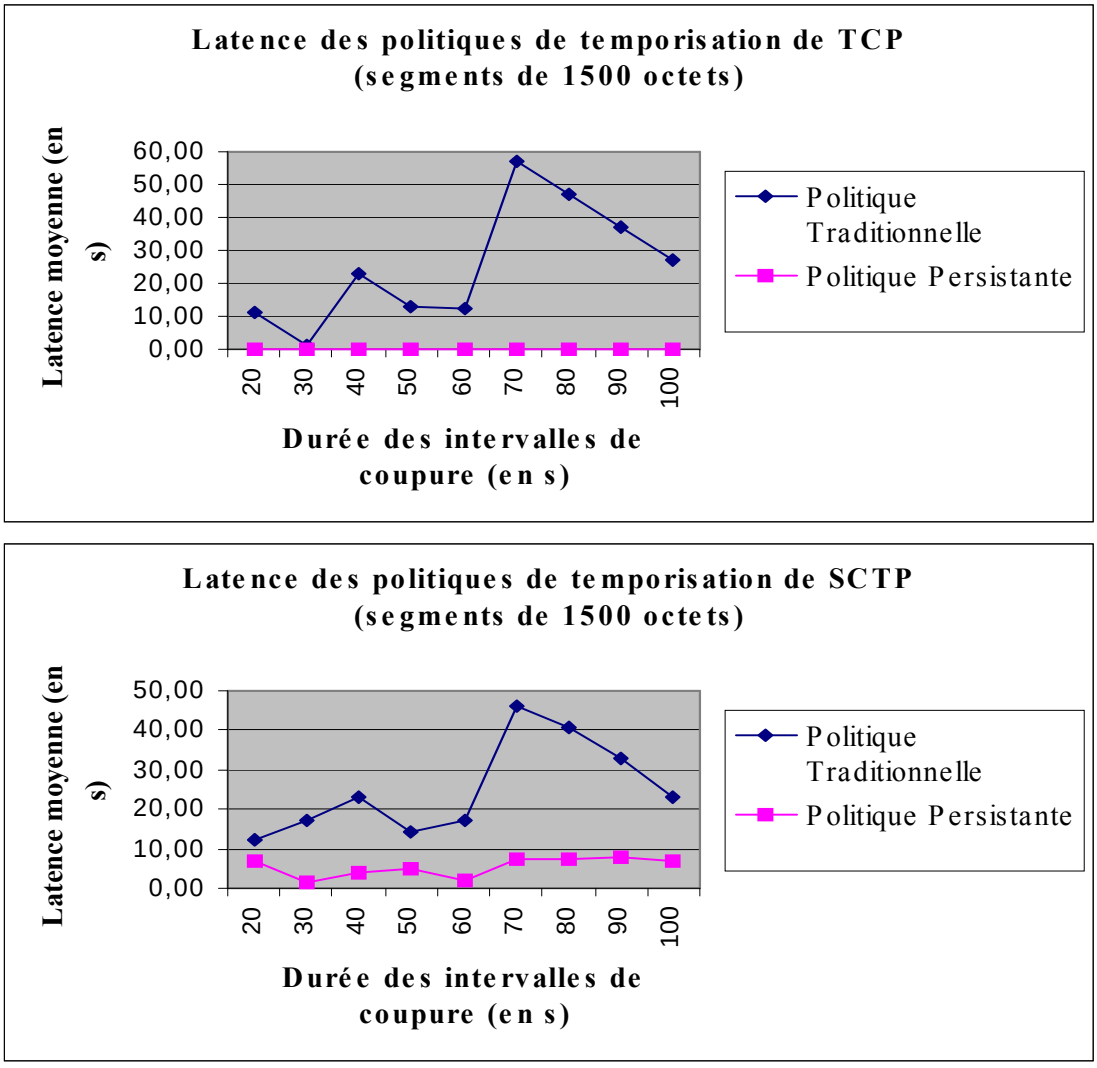


Figure IV.2. Evolution comparée des latences moyennes des politiques de temporisation de TCP et SCTP.

IV.5.4. Energie consommée par les émissions infructueuses

Le back–off exponentiel se traduit par des tentatives d'émission à chaque expiration du temporisateur d'envoi. Pour limiter le nombre de ces tentatives, le principe utilisé consiste à faire évoluer de façon exponentielle le temps d'attente avant retransmission entre deux tentatives consécutives. Dans le cas de l'indisponibilité due à un mauvais état du canal, la politique traditionnelle de retransmission utilisée engendre des ré–émissions de messages qui s'avèrent infructueuses. Or ces émissions infructueuses ont une influence notoire sur la consommation d'énergie, puisque les nœuds mobiles disposent d'une source d'énergie finie. Le protocole TCP ré–émet les segments de données à chaque tentative tandis que le protocole SCTP rajoute à l'émission infructueuse de ces messages de données, l'émission tout aussi infructueuse des messages Heartbeat de signalisation.

Le protocole SCTP étend la politique de retransmission traditionnelle telle que décrite pour les segments de données, dans le cas d'un nœud multi–domicilié. En plus de cette politique dans laquelle SCTP retransmet le message de données vers une adresse de rechange sur expiration du temporisateur d'envoi, le message Heartbeat est envoyé immédiatement à la destination ayant engendré l'expiration du temporisateur. En temps normal, les messages Heartbeat supplémentaires offrent un mécanisme à l'expéditeur pour mettre à jour plus fréquemment l'estimation du RTT de la destination de rechange, ce qui donne une meilleure estimation du RTT sur laquelle reposera la valeur du RTO.

Le processus de temporisation traditionnelle du protocole SCTP est donc différent de celui de TCP. Par exemple, lorsqu'un message est perdu lors de sa transmission vers sa destination première, il est retransmis plus tard vers la destination de rechange. Si le temporisateur de retransmission de cette dernière destination expire, le message perdu est retransmis une fois de plus vers une autre destination de rechange si elle existe, sinon, vers la destination primaire. En plus de ces multiples envois de messages de données vers ces différentes adresses, un message Hearbeat est également envoyé à la destination qui a engendré l'expiration du temporisateur. Par ce mécanisme, le protocole SCTP provoque plus de tentatives infructueuses de retransmission que TCP lorsque le mauvais état du canal se prolonge, en particulier dans le cas que nous avons choisi.

Les résultats observés sur les courbes de la figure IV.4 indiquent pour chaque protocole TCP et SCTP pris isolement, comme attendu, l'avantage de la politique persistante de retransmission sur la politique traditionnelle, en terme de réduction de la consommation d'énergie liée au nombre de tentatives infructueuses de retransmission. L'analyse croisée des résultats permet d'apprécier le comportement de chaque protocole. De part la différence de fonctionnement précédemment énoncée, l'avantage attendu dans le sens de la réduction du nombre de tentatives infructueuses est en faveur de TCP au détriment de SCTP pour chacune des deux politiques de retransmission. Un des inconvénients du mécanisme de détection de la défaillance de chemin du protocole SCTP est que les messages Heartbeat sont régulièrement émis même lorsque le canal est indisponible puisque cette information provenant de la couche liaison est inconnue de ce mécanisme de détection. Comme énoncé précédemment, l'indisponibilité du canal qui engendre l'absence d'acquittement impose à SCTP le recours à des retransmissions de messages de données (qui s'avèrent infructueuses) à chaque expiration de son temporisateur de retransmission.

Pour limiter le déséquilibre en défaveur du protocole SCTP du fait de l'émission des messages Heartbeat à chaque expiration du temporisateur, la comparaison entre les deux protocoles pour chaque politique de retransmission fixée est faite sur la base des émissions de messages de données sur les courbes de la figure IV.4. Comme le protocole SCTP est

handicapé par l'émission de messages Heartbeat, il émet moins de messages de données que le protocole TCP en politique traditionnelle. En poursuivant la comparaison des deux protocoles en politique persistante, le nombre d'émissions infructueuses reste le même, étant donné qu'aucun paquet de données n'est envoyé par l'un ou l'autre des deux protocoles durant l'indisponibilité du canal, pas même les messages Heartbeat de SCTP dans cette option de la simulation. La considération des émissions infructueuses sans tenir compte du type de message démontre une plus grande consommation d'énergie du protocole SCTP en politique persistante et en politique traditionnelle.

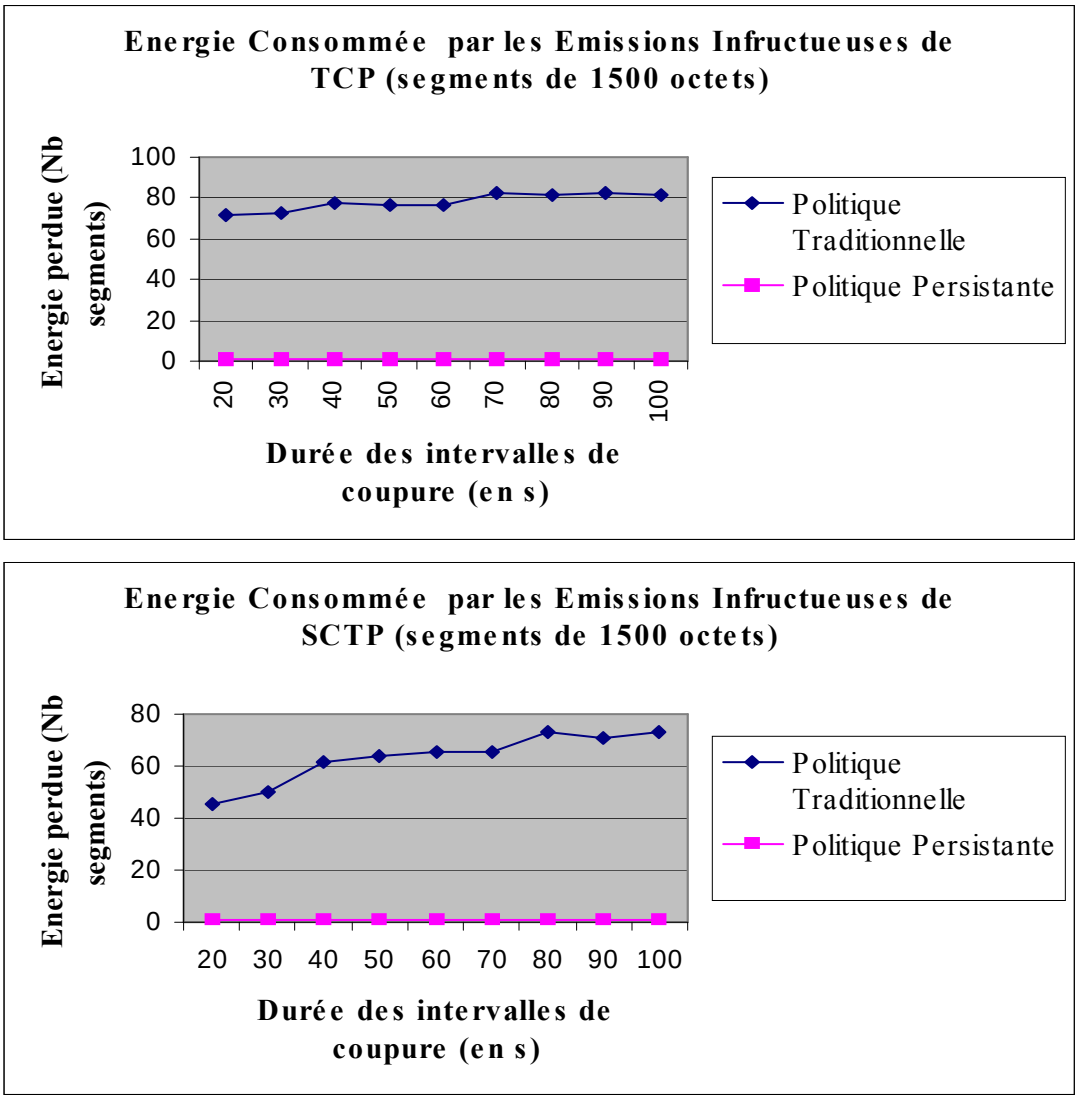


Figure IV.3. Evolution comparée des tentatives infructueuses d'émission des protocoles TCP et SCTP en fonction des politiques de temporisation.

IV.5.5. Pourcentage des émissions infructueuses par rapport aux données transférées

Les émissions infructueuses font partie du mécanisme de retransmission utilisé par les deux protocoles TCP et SCTP pour assurer une transmission fiable des données. Ces émissions infructueuses sont engendrées par le prolongement du mauvais état du canal, matérialisé ici par le modèle de coupure alterné. Les courbes de la figure IV.4 ci–dessus démontrent que le protocole TCP procède à plus d'émission infructueuses de paquets de données comparativement à SCTP soumis au mécanisme d'émission des messages de signalisation. Il est de ce fait intéressant d'évaluer le rapport entre le nombre de tentatives infructueuses et le nombre total de données transférées, pour mieux apprécier l'efficacité de chaque protocole. Les tentatives infructueuses, bien que consommatrices d'énergie, ne sont justifiées que pour assurer un transfert fiable de l'ensemble des données. Il peut être intéressant, sous l'angle de la comparaison des deux protocoles, de trouver le juste milieu entre le nombre de tentatives infructueuses et la quantité de données transférée, notamment dans le cas de la politique traditionnelle, puisque la politique persistante améliore le nombre de segments envoyés en offrant moins de tentatives infructueuses et donc moins de consommation d'énergie.

L'analyse croisée des résultats de la simulation donnés par les courbes de la figure IV.5 permet d'apprécier les performances de chaque protocole lorsque le nombre des émissions infructueuses est ramené en pourcentage des messages de données reçues à destination. En considérant ce calcul du taux d'émissions infructueuses consommatrices d'énergie en fonction de la quantité de données transférées, le protocole TCP est nettement favorisé par son débit supérieur à celui du protocole SCTP qui a un débit plus faible. Par exemple, dans le cas de la temporisation traditionnelle, pour des intervalles de coupure alternée de 100 secondes, le protocole TCP engendre moins de 0,6% d'émissions infructueuses par rapport au volume de données transférées tandis que le protocole SCTP avoisine 1,25%. Les courbes des politiques persistantes démontrent également l'avantage en faveur de TCP malgré l'utilisation des résultats de SCTP sans la plus grande partie des émissions infructueuses consacrée à l'émission des massages de signalisation.

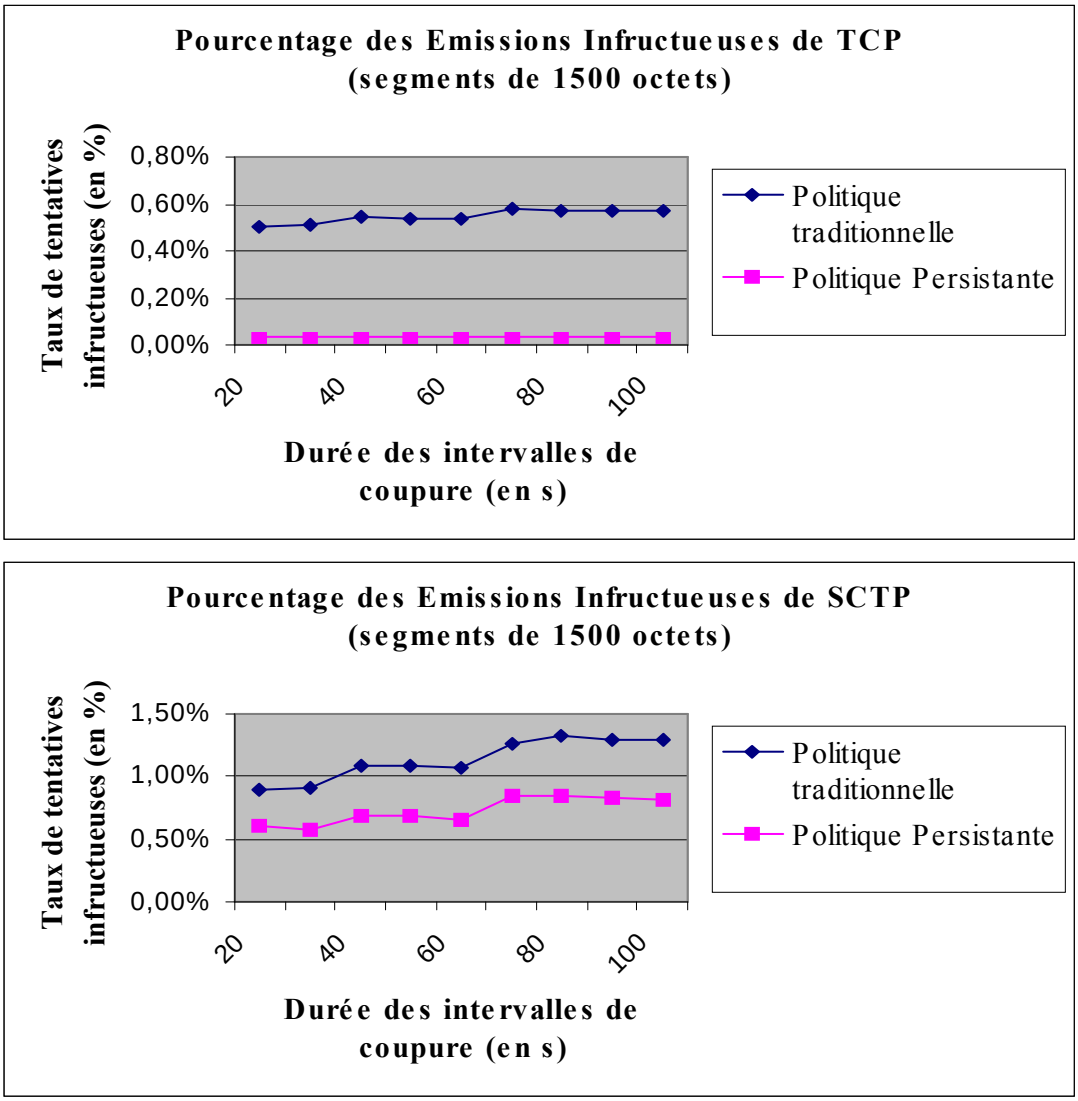


Figure IV.4. Pourcentage des Emissions infructueuses par rapport au volume de données transférées par les protocoles TCP et SCTP

IV.6. Conclusion

Le mécanisme de gestion d'association du protocole SCTP auquel est associé le mécanisme de détection de perte de chemin dans le réseau, par envoi de messages périodiques, constituent quelques points de différence de comportement qui existent avec le protocole TCP. L'application de la méthode RCL a explicité ces points de différence de fonctionnement et a permis de produire de nouveaux modèles conceptuels cross–layer. Ces modèles conceptuels ont reflété une convergence de réaction des deux protocoles aux interactions identifiées dans la deuxième étape de la méthode. Il a été alors possible d'étendre l'application de la temporisation persistante au protocole SCTP avec un canal à état variable. La temporisation persistante est une solution de rechange efficace au back–off exponentiel qui est décrit par le mécanisme de retransmission des deux protocoles, lorsque l'état du canal est mauvais. Elle repose sur l'information d'état du canal fournie par la couche liaison via les paramètres du sous–système environnement et les modules de notification explicite qu'il comporte.

Les résultats obtenus ont montré l'avantage de la temporisation persistante lorsque le nœud mobile est soumis aux aléas de la variation dynamique de l'état du canal. Cet avantage est traduit en terme de latence, de débit et de consommation d'énergie du fait des tentatives infructueuses de retransmission. Les résultats des simulations obtenus dans le cas de la comparaison des deux politiques traditionnelle et persistante de retransmission permettent d'apprécier les gains de la modification que nous proposons. La comparaison croisée des résultats obtenus pour chacun des protocoles reflète l'efficacité des mécanismes des deux protocoles TCP et SCTP qui constituent leurs principaux points de différence. Bien que ces résultats s'inscrivent dans le contexte de la comparaison des performances du mécanisme de temporisation utilisé par les deux protocoles pour assurer une transmission fiable, ils permettent d'apprécier de façon intuitive l'efficacité de chaque mécanisme. Le modèle de coupure alternée est utilisé dans les simulations pour décrire le canal sans fil. La politique persistante appliquée dans ce contexte au protocole SCTP remplace l'émission de messages Heartbeat destinés à établir l'accessibilité du nœud de destination. En procédant à ce remplacement, il est possible de trouver de nombreux cas de figure pour lesquels le protocole SCTP offre de meilleurs critères de performance par rapport à TCP, notamment pour les applications exigeantes en terme de fiabilité mais flexibles en terme d'ordre de séquence des messages [DIA03]. Comme ces protocoles s'inscrivent dans une pile complète, il est intéressant d'étudier le comportement des autres protocoles pour proposer des modèles cross–layer qui évitent la duplication des mécanismes d'une couche à une autre. C'est l'un des points qui sera abordé dans le prochain chapitre.

Un autre aspect important concerne la continuité de l'évaluation de l'état du canal, puisque la temporisation persistante y est intimement liée. Pour permettre au système de fournir une information à jour sur l'état du canal et servir ainsi la temporisation persistante, le fonctionnement global doit être étudié et au besoin des mécanismes complémentaires seront proposés. C'est le deuxième aspect abordé dans le chapitre suivant.

Chapitre V. Apport du routage à la temporisation persistante

V.1. Introduction

La temporisation persistante est un mécanisme que nous avons proposé pour améliorer les temporisations des protocoles fiables de la couche transport. Elle repose sur le principe cross–layer d'évaluation et de mise à disposition de l'état du canal par la couche liaison. L'utilisation de cette temporisation dans la pile complète de protocoles permet de mettre en évidence les mécanismes complémentaires nécessaires au fonctionnement normal du système.

De façon générale, le fonctionnement normal des mécanismes fondés sur les paramètres d'état du canal dépend de la continuité de l'évaluation de ces paramètres. Cette continuité d'évaluation est donc un principe clé à prendre en compte pour la mise en œuvre de la temporisation persistante, sous la contrainte d'éviter d'amoindrir l'apport de cette temporisation en terme de performances, qui ont été évaluées dans les précédentes simulations.

Le comportement persistant des protocoles fiables de la couche transport lors du mauvais état du canal a la particularité d'engendrer un silence des émissions de ces protocoles. L'activité étant ainsi réduite, le comportement des autres couches et de l'ensemble du système doit être pris en compte pour assurer la continuité de l'évaluation de l'état du canal. Par exemple, si l'un des protocoles de routage pro–actif est utilisé au niveau réseau, l'émission périodique des paquets de signalisation est un mécanisme complémentaire favorable au principe de la continuité d'évaluation. C'est pourquoi nous consacrons un point de ce chapitre à l'étude des protocoles de routage ainsi qu'à la problématique de la continuité de l'évaluation.

Pour mettre en œuvre le principe de l'évaluation continue de l'état du canal en considérant les mécanismes du système global, l'analyse des combinaisons possibles de la catégorie des protocoles disponibles aux niveaux transport et réseau permet d'expliciter le travail conceptuel complémentaire à fournir. Au niveau transport nous considérons la catégorie des protocoles fiables, tandis qu'au niveau réseau, le comportement du système cross–layer est considéré à travers l'utilisation de l'un des protocoles de routage pro–actif ou ré–actif. Cette analyse rend possible l'observation de la redondance d'émission des paquets de détection et de signalisation lorsque par exemple le protocole SCTP, qui émet régulièrement des paquets "HeartBeat" pour la gestion des chemins, est associé à un protocole pro–actif comme OLSR qui émet des messages "Hello" à une fréquence donnée pour établir la topologie du réseau. Il est tout aussi possible d'observer la duplication de la temporisation d'attente d'envoi, même à des échelles de temps différentes, entre la couche transport et la couche réseau lorsque l'un des protocoles TCP ou SCTP est utilisé en combinaison avec un protocole de routage ré–actif.

Après l'analyse des deux protocoles fiables de la couche transport dans les précédents chapitres, nous avons choisi de mener une étude comparative des protocoles de routage du réseau ad–hoc dans ce chapitre du fait de l'importance de la combinaison de protocoles utilisés d'une part au niveau de la couche transport pour un transfert fiable et d'autre part au niveau réseau pour le routage de paquets. Cette importance de la combinaison "protocole de transport fiable – protocole de routage" se traduit dans les performances du système global en général et dans celle du système cross–layer que nous proposons en particulier. Le but final de cette étude comparative des protocoles de routage sera de compléter la description des modèles cross–layer proposés et de rendre les performances mesurées indépendantes de la combinaison des protocoles transports fiables et protocoles de routage utilisés. Les acquis de l'architecture nécessitent d'assurer une telle indépendance. Elle permettra non seulement d'assurer un fonctionnement normal et transparent du système cross–layer proposé, mais elle permettra également d'assurer l'évolution modulaire du système, la modularité étant un des avantages de l'architecture.

La politique de temporisation persistante proposée se rapporte aux protocoles fiables de la couche transport. Le présent chapitre est consacré à l'analyse de l'apport du routage à cette temporisation du fait de la nécessité de l'évaluation continue de l'état du canal. C'est pourquoi une autre attente de l'étude comparative des protocoles de routage sera de mettre en exergue la connaissance des points communs et des points de différence entre ces protocoles, pour mettre en place le nécessaire principe d'indépendance entre les couches transport et réseau du système cross–layer proposé.

V.2. Problématique de l'utilisation de l'état du canal dans la politique de temporisation persistante

Le principe de base de la temporisation persistante repose sur la mise à disposition continue de l'état à jour du canal sans fil par la couche liaison via le sous–système environnement. Dans les modèles cross–layer précédents, l'état du canal est reflété par les paramètres SNR, BER, taux de perte des paquets envoyés ou reçus, taux de retransmission, tous ces paramètres étant calculés à partir de l'activité de la couche physique.

En référence à la temporisation persistante, le prolongement du mauvais état du canal entraîne un comportement silencieux de la couche transport. Ce silence implique une forte réduction de l'activité du nœud, et cette activité sera liée pendant cette période, à une probable activité des couches inférieures. Or, l'évaluation continue de l'état du canal est nécessaire, pour permettre au nœud de sortir au moment opportun, de l'état silencieux engendré par la persistance et lui permettre ainsi d'améliorer sa latence. Pour la poursuite de l'évaluation de l'état du canal, le nœud dispose de plusieurs niveaux d'événements. Il y a d'une part l'activité extérieure au nœud qui est une source potentielle et que la modélisation doit prendre en compte. Cette prise en compte se fera par l'indication de la date de mise à jour de ces paramètres et le jugement de son caractère récent ou non. D'autre part, une autre source potentielle de continuité de l'évaluation de l'état du canal provient de l'activité latente ou régulière des couches inférieures.

Une réponse possible à la contrainte de l'évaluation continue consiste à assurer l'émission par le sous–système environnement des paquets allégés de signalisation. Le besoin de découverte du voisinage immédiat du nœud persistant pour assurer la connexion au réseau peut justifier une telle émission. Ces paquets ne feront pas l'objet de rediffusion, leur émission ne sera pas systématique et périodique mais contrôlée et se fera lorsqu'une demande explicite

est adressée au sous–système environnement par les protocoles transport et réseau. Un nœud voisin n'a pas l'obligation d'y répondre à moins d'en être le destinataire. Cette solution nécessite de définir un nouveau protocole et la nature des messages qu'il utilise. C'est pourquoi nous avons préféré une solution plus simple. En effet, en analysant le comportement global du réseau en général et de la pile des protocoles du nœud persistant en particulier, il est possible d'optimiser la mise en place du mécanisme qui doit garantir la continuité de l'évaluation de l'état du canal sans fil. Plusieurs cas possibles peuvent être distingués.

L'activité ambiante liée à l'état de forte charge du réseau permet de répondre au besoin de continuité de l'évaluation. En effet, dans un tel contexte, il y a suffisamment d'activité RTS/CTS ou de transmission de paquets dans le voisinage du nœud pour permettre de calculer les valeurs des paramètres d'état du canal, sans overhead, sans tenir compte de la destination des paquets. La mise à jour des dates de calcul de ces paramètres permettra de savoir si le système doit activer un mécanisme de compensation ou schéma auxiliaire. Cette activation du mécanisme de compensation est contrôlée par des dates de calcul jugées trop vieilles.

Contrairement à l'état de forte charge, lorsque le réseau est dans un état de charge légère, le canal est plus souvent libre. Avant de déclencher le mécanisme de compensation, il est nécessaire de prendre en compte les mécanismes qui existent au sein des couches inférieures et qui permettent d'assurer la continuité du calcul des paramètres d'état du canal. Par exemple, la continuité peut être assurée lorsque le protocole de routage utilisé émet périodiquement ou sur perte de connectivité (comme le cas d'un mauvais état du canal) des messages de contrôle pour établir la connectivité du réseau (protocole pro–actif). Dans ce cas, le schéma auxiliaire n'est pas nécessaire, les paquets émis par la couche de routage permettront aux couches liaison et physique de procéder à l'évaluation continue.

Dans le cas des protocoles de routage ré–actifs par exemple qui n'émettent pas de messages d'établissement de la topologie du réseau, le mécanisme de compensation va consister à ce que le sous–système environnement déclenche une demande de mise à jour de la route menant à des destinations particulières dont l'une des destinations référencées dans les paquets en instance d'envoi. La demande sera simplement adressée au protocole de routage. L'avantage de ce schéma auxiliaire est qu'il n'est pas nécessaire de définir un protocole supplémentaire, mais seulement d'ajouter un module au sous–système environnement pour assurer cette fonctionnalité. L'activité de ce module se fera selon le même principe de partage des ressources réseaux utilisé par la diffusion des messages périodiques. En adoptant ce scénario, l'utilisation de la politique persistante à faible charge de réseau engendrera seulement une requête de mise à jour de route pour l'une des adresses de destination des paquets en cours d'envoi et permettra ainsi d'assurer la continuité de l'évaluation de l'état du canal.

Dans le point suivant, nous étudions la faisabilité des deux solutions liées aux protocoles de routage. Nous menons une étude du comportement des protocoles de routage appartenant aux deux familles qui sont la famille des protocoles pro–actifs et celle des protocoles ré–actifs.

V.3. Etude du routage dans les réseaux Ad–hoc

La problématique du routage dans les réseaux ad–hoc a donné lieu à une effervescence sans précédent dans le domaine réseau. Notre objectif n'est donc pas de faire une présentation exhaustive de toutes les solutions présentées, ni même de les comparer. Nous ne les étudions qu'au travers des besoins évoqués précédemment afin d'étayer nos propositions.

Le routage est le mécanisme qui assure l'acheminement de l'information à travers le réseau. La couche réseau est utilisée pour assurer cette fonction dans les réseaux MANET. Le routage d'information des réseaux sans fil découle de l'adaptation qui est faite des protocoles du réseau filaire. En effet, les réseaux filaires utilisent les protocoles de routage pour établir les routes qui mènent d'une station à une autre. Ces routes sont établies à l'avance dans des tables de routage, sur la base de l'échange périodique d'information entre les stations. Ce principe d'établissement des routes à l'avance a conditionné le regroupement de ces protocoles dans la catégorie des protocoles pro–actifs. Leur algorithme de fonctionnement détermine leur classification en protocoles fondés sur l'algorithme de vecteur de distance et en protocoles fondés sur l'algorithme à état de liens. Ils appartiennent tous à la classe des protocoles à plus court chemin et à routage distribué.

Les protocoles de routage des réseaux ad–hoc ont des notions similaires à celles des réseaux filaires. Les protocoles pro–actifs acheminent des paquets périodiques de contrôle à travers ce réseau sans fil pour mettre à jour la connaissance qu'ils ont de la topologie à un instant donné. Les travaux actuels qui portent sur cette famille de protocoles concernent le protocole DSDV (Destination Sequenced Distant Vector) [PER01], le protocole OLSR (Optimised Link State Routing) [JAC01] [LAO98], le protocole FSR (Fisheye State Routing) [GER01] qui sont soumis au groupe MANET de l'IETF [RFC2501] [MAN]. Théoriquement, avant d'expliciter le comportement de chacun des protocoles de cette famille, leur fonctionnement a l'avantage d'assurer le principe de l'évaluation continue de l'état du canal sans fil, que le réseau soit à forte charge ou à charge légère.

Du fait que les réseaux sans fil sont moins performants que les réseaux filaires en terme de débit, de nouvelles techniques de signalisation et de routage ont été conçues pour améliorer l'utilisation de la bande passante. Les réseaux sans fil sont caractérisés par une topologie dynamique. La prise en compte de cette caractéristique dynamique a amené à la mise en place des techniques utilisées à la demande, de telle sorte que la route est installée lorsque le nœud décide d'envoyer une information vers un autre nœud. Dans plusieurs cas, ce principe permet de résoudre le problème de l'expiration de la route installée à l'avance lorsque la topologie du réseau change. La famille de protocoles ré–actifs est composée des protocoles qui établissent les routes à la demande, lorsque les applications en ont besoin, par l'envoi d'une requête de route à travers le réseau afin d'obtenir la route qui mène à la destination recherchée. Cette famille regroupe les protocoles suivants soumis à l'étude du groupe MANET de l'IETF : le protocole AODV (Ad–hoc On Distance Vector) [ROY00] [DAS01] [ROY99], le protocole DSR (Dynamic Source Routing), le protocole TORA (Temporally Ordered Routing Algorithm) [COR01] [TOR97], le protocole ABR (Associativity Based Routing) [ZIN01]. L'utilisation de l'un des protocoles de cette famille doit normalement nécessiter le déclenchement du mécanisme de compensation, lorsque le réseau se trouve dans un état de fonctionnement à charge légère.

V.3.1. Comparaison des protocoles de routage ad–hoc

Les deux tableaux ci–dessous comparent quelques protocoles de routage sur la base de critères particuliers, dont le critère relatif à l'émission de message périodique. Dans le cadre de la continuité de l'évaluation de l'état du canal, ce critère donne une indication de l'activation qui doit être faite ou non du mécanisme de compensation pour assurer la continuité de l'évaluation lorsque le réseau fonctionne à faible charge. Les protocoles qui émettent des messages périodiques doivent permettre au module du sous–système environnement d'économiser l'activation du mécanisme de compensation. Les autres critères concernent essentiellement la fonctionnalité de routage que ces protocoles doivent assurer avec l'efficacité attendue.

Critère de comparaison	**AODV**	**DSR**	**OLSR**	**FSR**
Sans boucle	Oui	Oui	Oui	Oui
Plusieurs routes possibles	Non	Oui	Non	Non
Distribué	Oui	Oui	Oui	Oui
Type	Ré–actif	Ré–actif	Ré–actif	Pro–actif
Messages périodiques de contrôle	Non	Non	Oui	Oui
Liens unidirectionnels	Non	Oui	Oui	Oui

Critère de comparaison	**CBRP**	**LANMAR**	**TBRPF**	**ZRP**
Sans boucle	Oui	Oui	Oui	Oui
Plusieurs routes possibles	Oui	Non	Non	Oui
Distribué	Oui	Oui	Oui	Oui
Type	Hybride	Hybride	Pro–actif	hybride
Messages périodiques de contrôle	Oui	Oui	Oui	Oui
Liens unidirectionnels	Oui	Non	Non	Oui

Table V.1. Comparaison de quelques protocoles de routage en réseau ad–hoc

Légende :
CBRP : Cluster Based Routing Protocol [JIA99]
LANMAR : Landmark Routing Protocol [PEI00]
ZRP : Zone Routing Protocol [HAA98]
TBRPF : Topology Broadcast Based on Reverse – Path Forwarding [RFC3684]

Chaque protocole du tableau ci–dessus dispose de sa propre philosophie de fonctionnement. Le protocole DSR et le protocole DSDV sont deux exemples de protocoles de routage ad–hoc appartenant le premier à la famille des protocoles ré–actifs et le second à la famille des protocoles pro–actifs.

Le protocole DSR est un protocole à routage d'information dynamique fondé sur la découverte de route et la maintenance de route. La découverte de route est le mécanisme qu'utilise le protocole lorsque le nœud dispose d'un paquet à envoyer vers un autre mais ne dispose pas de chemin qui mène à cette destination. Le nœud émetteur initie une requête de route, les nœuds intermédiaires qui ne disposent pas du chemin recherché inscrivent leur adresse dans le champ liste d'adresses, les nœuds intermédiaires disposant du chemin recherché ou le nœud de destination renvoient ce chemin complet à l'émetteur de la requête de route. Le mécanisme de maintenance de route est utilisé pour assurer plusieurs fonctions. Les fonctions réalisées se rapportent à l'utilisation des acquittements, aux messages d'erreur de route, à la récupération de paquets par la modification de la route proposée par la source du paquet et l'envoi du message d'erreur de route au nœud émetteur, et à la fragmentation du paquet pour ajuster sa taille au chemin.

Le protocole DSDV repose sur la méthode de Bellman Ford utilisée par le protocole RIP (Routing Information Protocol) du réseau filaire. Lorsque le protocole DSDV est utilisé pour le routage d'information, chaque nœud du réseau détient une table contenant les nœuds de destination, le nombre de sauts pour atteindre chaque destination, le prochain saut auquel le paquet doit être envoyé, le numéro de séquence et la dernière date de mise à jour. Ces tables sont régulièrement mises à jour et cette activité implique un overhead important. En plus de l'overhead qui constitue un réel problème, le protocole DSDV nécessite également une estimation de la mobilité pour déduire la fréquence de mise à jour des tables de routage, avec le handicap majeur de l'existence de peu de travaux autour de la mobilité. Enfin, les tailles des tables générées par le protocole constituent un autre problème non moins important pour les réseaux de grande taille.

Le choix du protocole de routage à utiliser dans le réseau ad–hoc reste ouvert, et dépend du critère à optimiser, chaque protocole ayant ses avantages et ses points faibles. Des évaluations comparatives des performances ont été réalisées pour mesurer le gain apporté par chaque protocole, comme présenté dans le paragraphe suivant. L'importance de cette partie dans la conception du système cross–layer global réside dans le fait que la connaissance du fonctionnement du protocole de routage utilisé au niveau de la couche réseau, rend possible l'assurance de l'évaluation continue de l'état du canal pour les besoins de la temporisation persistante des protocoles transports fiables. Le protocole de routage détermine le comportement du système cross–layer global à faible charge du réseau. Le système utilisera l'émission périodique des messages de contrôle ou des paquets de relance de découverte de route. Dans ce dernier cas, le paquet de relance à envoyer par le module du sous–système environnement peut simplement être le dernier. La connaissance du comportement des protocoles de routage détermine également la nature des propositions qui doivent être faites pour assurer le fonctionnement normal des mécanismes cross–layer.

V.3.2. Comparaison des performances des protocoles

Les deux études présentées sont utilisées dans plusieurs publications malgré leur date de parution assez anciennes, 1998 et 2000. La première étude procède à la comparaison des quatre protocoles de routage DSDV, TORA, DSR et AODV sur la base de critères simples. La seconde étude présentée utilise des critères de comparaison plus détaillés.

V.3.2.1. Etude comparative à critères d'évaluation simples

Dans le cadre du projet Monarch [BRO98], les auteurs Broch, D. Maltz, D. Johnson et Y. Hu ont réalisé une comparaison de performances des quatre protocoles DSDV (pro–actif), TORA, DSR, AODV (ré–actif s), dans l'environnement ns–2, en utilisant le modèle de mobilité "random waypoint" et des liens considérés comme bidirectionnels. Le contrôle de la charge du réseau est facilité par l'utilisation de sources de trafic de type CBR (Constant Bit Rate). La simulation de 900 secondes porte sur 50 nœuds mobiles avec un nombre de sources variable de l'ordre de 10, 20 et 30. Les critères d'évaluation suivants ont été utilisés :

- Le taux de perte : son importance se traduit dans la gestion de la transmission des données par la couche transport et peut de ce fait influencer le débit maximal que le réseau peut écouler.
- L'overhead de routage : ce paramètre doit être aussi faible que possible pour permettre d'optimiser la bande passante du réseau. il est mesuré en terme de nombre de paquets.
- La pertinence de chemin : ce paramètre constitue la différence entre le chemin utilisé par les données et le plus court chemin qui existe entre l'expéditeur et le récepteur. La pertinence du chemin montre la capacité du protocole à découvrir des routes efficaces en terme de nombre de nœuds intermédiaires.

Le temps d'arrêt qui est un paramètre en entrée du modèle de mobilité détermine le niveau de mobilité choisi dans la comparaison des quatre protocoles. Les protocoles AODV et DSR sont plus efficaces en terme de taux de perte quelque soit le nombre de sources. Pour le critère relatif au contrôle de trafic, le protocole DSR est largement supérieur aux trois autres protocoles. En exprimant l'overhead en terme de nombre de paquets, l'avantage revient au protocole DSR, mais son entête comporte une taille substantielle du fait qu'il contient la route que le paquet doit emprunter. Si cette charge est prise en compte, le protocole AODV devient meilleur excepté pour une forte mobilité traduite par un temps d'arrêt trop court. Les protocoles DSDV et DSR sont plus efficaces en terme de pertinence de chemin. Contrairement aux protocoles TORA et AODV, ce critère d'évaluation est relativement indépendant du degré de mobilité. Les résultats globaux sont favorables au protocole DSR et de façon moindre au protocole AODV, qui constituent les seuls protocoles en voie de normalisation.

Les résultats obtenus dans la simulation décrite ci–dessus doivent être reconsidérés pour diverses raisons dont le degré de mobilité choisi qui n'est pas nécessairement représentatif de la dynamique du réseau. De même, le contrôle de trafic doit aussi être comparé au trafic de données acheminé par chaque protocole. Un protocole peut générer plus d'overhead et acheminer plus de données. Le ratio de contrôle de volume par rapport au volume de données est plus judicieux [PER01]. Le critère de distance pour définir la pertinence de chemin est suffisamment réducteur, les résultats en terme de délai ne sont pas présentés.

V.3.2.2. Etude comparative à critères d'évaluation plus détaillés

Des travaux plus récents menés par les auteurs S. Das, C. Perkins et E. Royer permettent de comparer les protocoles AODV et DSR avec plus de détails [PER01]. L'environnement ns–2 est utilisé comme dans les précédentes simulations, avec des sources de type CBR et un modèle de mobilité "random waypoint". Les scénarios utilisent deux surfaces de simulations : 1500 x 300 avec 50 nœuds et 2200 x 600 avec 100 nœuds. Le nombre de sources varient de 10 à 40, tous les liens sont considérés comme bi directionnels. Les critères de performance évalués sont le taux de perte, le délai de bout en bout et le ratio de l'overhead par rapport à l'information (en nombre de paquets respectifs). L'estimateur de la mobilité est toujours le temps de pause.

Les résultats obtenus à partir du test des deux protocoles dans la configuration de 50 nœuds ont permis de fixer les valeurs des critères d'évaluation. Le taux de perte de paquets des deux protocoles avec 10 et 20 sources sont équivalents. Le protocole AODV prend de l'avantage avec plus de sources (30 ou 40). Le protocole DSR perd entre 30 et 50% de paquets en plus que le protocole AODV à forte mobilité (temps de pause trop court).

Le délai de bout en bout donne un avantage de rapidité en faveur du protocole DSR qui devient 4 fois plus rapide que le protocole AODV pour 10 sources, de même pour 20 sources mais avec une différence moindre. Au delà de 30 sources, la tendance est inversée, AODV devient deux fois plus rapide et affiche des meilleurs performances en particulier avec un temps de pause très court.

Avec le ratio overhead sur données, le protocole DSR est meilleur dans tous les cas avec un facteur d'au moins 4. Lorsque le nombre de sources augmente, le protocole DSR garde un rapport relativement constant par rapport au protocole AODV, même lorsque les autres critères deviennent défavorables. Le délai des deux protocoles est très long avec 40 sources, même avec une faible mobilité. Ceci s'explique par l'absence de mécanismes de partage de charge et par la politique de sélection des routes. Par exemple pour le protocole DSR, le plus court chemin en terme de nombre de sauts est constamment considéré comme le meilleur chemin.

Avec 100 nœuds mobiles, les résultats du taux de perte et du délai de bout en bout sont identiques à ceux du scénario avec 50 nœuds. La différence entre l'overhead et le ratio des données est plus faible, avec de plus un rapport instable pour le protocole DSR.

Le protocole AODV est meilleur que le protocole DSR si le nombre de sources augmente en terme de délai et de taux de perte. Même si l'overhead du protocole DSR reste meilleur, ces valeurs ont été calculées en nombre de paquets sans tenir compte de leur taille. Le comportement peut être expliqué par le fait que dans les deux réseaux ad–hoc, l'accès au médium est plus coûteux que l'ajout d'octets supplémentaires à un paquet existant.

Le comportement des deux protocoles est considéré à charge variable, de 0 à 800 kbits/s et en considérant seulement 10 sources. Les critères considérés sont le délai de bout en bout, le débit de réception et l'overhead (en kbits/s). le débit du protocole DSR est saturé à partir de 325kbits/s du fait du taux de perte élevé tandis que le seuil du protocole AODV est à 700 kbits/s. Le délai du protocole DSR est meilleur avec une charge faible (au moins par un facteur de 2), mais beaucoup plus long à forte charge. L'overhead des deux protocoles est légèrement en faveur du protocole DSR. Dans le scénario comportant 40 sources, le comportement des deux protocoles est le même, mais les seuils de saturation sont plus faibles, 150 pour le protocole DSR et 300 pour le protocole AODV. De même, l'overhead des deux protocoles devient plus grand que le débit de réception.

L'étude montre les réactions, les avantages et les contre–performances des protocoles de routage comparés, en particulier ceux du protocole DSR lorsque le nombre de sources et la charge du réseau augmentent. L'avantage de cette étude est de permettre de faire un choix du protocole de routage à utiliser dans un réseau ad–hoc, en fonction du critère d'optimisation choisi. Notre proposition de modèles conceptuels cross–layer tient compte de ces limites que sont les valeurs des critères de performance. Ces limites doivent être au minimum assurées et renforcées pour justifier la mise en place de ces mécanismes cross–layer. Chaque protocole de routage a sa particularité de fonctionnement. Ces protocoles peuvent concourir à la définition de mécanismes cross–layer complémentaires à la temporisation persistante. La continuité de l'évaluation est un exemple, mais d'autres mécanismes propres à la fonction de routage peuvent être conçus pour que le choix du protocole de routage soit totalement transparent au fonctionnement des mécanismes cross–layer lorsqu'ils sont implantés. Pour déterminer l'implication sur les modèles conceptuels cross–layer proposés, lorsqu'un protocole de routage est remplacé par un autre dans la pile de protocoles, nous procédons dans le point suivant à l'application des 3 premières étapes de la méthode RCL à des protocoles de routage choisis dans les deux familles.

V.4. Modèles conceptuels cross–layer des protocoles de routage

L'application de la méthode RCL au chapitre II et les modèles conceptuels cross–layer qui ont été produits, ont révélé les apports du protocole de routage utilisé par la couche réseau (protocole DSR). Ces contributions du protocole DSR dans la définition des modèles conceptuels cross–layer se traduisent en terme de génération d'AACL et de leur utilisation. Dans cette partie de ce chapitre, nous utilisons les deux principes qui définissent ces contributions, à savoir la génération et l'utilisation des AACL par les protocoles de routage, afin de dégager les points communs et les points de différence entre les protocoles des deux familles. L'objectif est de mesurer l'impact d'un changement de protocole de routage sur les modèles cross–layer, et de ce fait, proposer un ou plusieurs mécanismes cross–layer complémentaires au mécanisme de la temporisation persistante.

La première orientation consiste à comparer les AACL générées par les protocoles de routage afin de retrouver les similitudes et les points de discordance entre ces protocoles. Ainsi, nous limitons l'utilisation de la méthode RCL à ses trois premières étapes qui sont : le choix de la pile de protocole, le recensement des AACL et la production du tableau d'interaction des protocoles. Les étapes de choix de la pile des protocoles et de recensement des AACL seront simplifiées du fait que les modifications qui seront apportées à la pile de protocoles utilisée au chapitre II sont réduites au remplacement du protocole de routage. Le recensement des AACL se fera en utilisant le même principe que celui utilisé au chapitre IV, lors du remplacement du protocole TCP par le protocole SCTP. Les quatre protocoles de routage suivants feront l'objet de cette étude : le protocole OLSR et le protocole DSDV de la famille du routage pro–actif, et le protocole AODV de la famille du routage ré–actif, en plus du protocole DSR déjà utilisé. Pour uniformiser les recensements et conserver leur caractère exhaustif pour les besoins de la comparaison, nous complétons les AACL générées par le protocole DSR en ajoutant la mise à disposition de la table de routage et la mise à disposition de la liste noire. Le tableau d'interaction des protocoles produit à la troisième étape de la méthode RCL permettra de synthétiser les apports de chaque protocole de routage et de déduire leurs AACL de convergence et de divergence.

V.4.1. Cas du protocole OLSR

V.4.1.1. Choix de la pile de protocoles

La pile de protocoles utilisée au chapitre II est légèrement modifiée. Les protocoles TCP, IP et 802.11 sont conservés, le routage est assuré par le protocole OLSR à la place du protocole DSR.

V.4.1.2. Recensement des AACL

En récapitulant les points clés de fonctionnement du protocole OLSR, nous recensons les AACL qu'il génère et dont la liste permet de définir sa contribution aux modèles conceptuels cross–layer. Ainsi, les interactions suivantes peuvent être recensées :

- L'AACL de "Mise à disposition" de la liste des nœuds MPR : OLSR est un protocole pro–actif à base de table. Le concept clé utilisé dans le protocole est la notion de relais multipoint (MPR pour MultiPoint Relay). Les MPR d'un nœud du réseau sont des nœuds qu'il sélectionne dans son voisinage immédiat bidirectionnel à 1 saut pour acheminer les messages qu'il diffuse durant le processus d'inondation (ou flooding). Cette technique d'utilisation des MPR réduit substantiellement l'overhead des messages comparativement au flooding classique dans lequel chaque nœud retransmet le message dont il a reçu copie. Le lien bidirectionnel (symétrique) permet d'éviter le problème de liens asymétriques lors du transfert des données tel que le problème d'impossibilité d'obtenir les acquittements de la couche liaison sur les liens unidirectionnels. Le MPR est choisi parmi les voisins bidirectionnels à 1 saut de telle sorte qu'il puisse acheminer les informations du nœud qui l'a élu à destination des nœuds à 2 sauts en terme de portée radio. De plus, avec le protocole OLSR, les informations d'état de lien sont générées uniquement par les MPR élus, ce qui constitue la seconde optimisation qui minimise le nombre de messages de contrôle injectés dans le réseau. OLSR requiert uniquement des informations partielles d'état de lien destinées à fournir des routes à plus court chemin. Une troisième optimisation du protocole OLSR est liée au fait qu'un nœud MPR peut choisir de donner des informations sur les liens uniquement aux nœuds qui l'ont élu. Par conséquent, contrairement à l'algorithme d'état de lien classique, une information partielle d'état de lien

est distribuée dans le réseau. Cette information est alors utilisée pour le calcul des routes. Les nœuds non MPR reçoivent et traitent les messages diffusés mais ne les rediffusent pas.

- L'AACL de "Mise à disposition" de la liste MPR Selector Set : Chaque nœud maintient la liste des nœuds qui l'ont élu comme MPR, cette liste est appelée MPR Selector Set. Le nœud obtient cette information à partir des messages Hello qu'il reçoit de ses voisins.
- L'AACL de "Mise à disposition" de la liste des nœuds voisins à 1 saut : la fonction de détection des voisins est assurée par le protocole OLSR dans un réseau de nœuds à interface unique. En effet, à partir de l'échange d'information, un nœud donné déduit l'ensemble des liens directs qui le lient aux autres nœuds du réseau. L'adresse principale d'un nœud à interface unique est par définition l'adresse de la seule interface de ce nœud. Dans un réseau de nœuds à interfaces multiples, des informations complémentaires sont nécessaires pour associer les adresses des interfaces aux adresses principales. Ces informations sont acquises à travers les messages MID de déclaration d'interfaces multiples. La notion de voisinage revêt une importance notoire dans le déroulement du protocole comme par exemple dans le traitement d'un message reçu qui passe par plusieurs étapes de vérification. Si l'adresse de l'interface de l'émetteur du message n'est pas détectée comme étant dans le voisinage symétrique à 1 saut du nœud, l'algorithme d'acheminement s'arrête à cette étape.
- L'AACL de "Mise à disposition" de la liste des nœuds accessibles : La table de routage maintenue par chaque nœud est calculée sur la base des informations contenues dans la "local link information base" et le "Topology Set". Pour chaque changement opéré au niveau de l'un de ces deux ensembles, la table de routage est recalculée et les routes menant à toutes les destinations du réseau sont ainsi mises à jour. Chaque entrée de la table de routage spécifie le fait que le nœud de destination X est estimé à D sauts à partir du nœud courant et que le nœud voisin symétrique qui est le prochain saut de la route est relié à l'interface I. Les entrées de la table de routage concernent toutes les destinations du réseau pour lesquelles une route est connue. Les destinations dont la route est coupée ou connue partiellement ne sont pas enregistrées. La table de routage est mise à jour lorsqu'un changement est détecté dans l'un quelconque de ensembles suivants : "Link Set", "Neighbor Set", "2-hop Neighbor Set", "Topology Set", "Multiple Interface Association Information Base". La mise à jour de ces ensembles ne génère pas de transmission de message, ni dans le réseau ni dans le voisinage à 1 saut. Nous appliquerons un simple filtre à la table de routage fournie par cette AACL. Les nœuds appartenant aux listes des autres AACL du protocole ne doivent pas figurer dans cette table pour éviter la redondance et rendre les listes complémentaires.
- L'AACL de "Mise à disposition" de la liste noire : tout nœud dont la communication bidirectionnelle ne peut être établie, par exemple par absence d'acquittement de la couche liaison caractéristique des liens unidirectionnels, est considéré comme appartenant à la liste noire. L'état du nœud est mis à jour dès qu'une nouvelle information confirme son accessibilité symétrique.

En plus des AACL recensées ci–dessus, le protocole OLSR utilise les informations cross–layer fournies par les autres couches. Les points suivants définissent la nature de l'utilisation qui est faite des interactions cross–layer provenant des autres couches.

- Utilisation de l'AACL de "Mise à disposition" du RSS de la couche liaison : le mécanisme de perception des liens du protocole OLSR se fait à travers l'émission périodique des messages "Hello" sur les interfaces actives. Un message "Hello" est généré pour chaque interface. Un ensemble local de liens est créé pour décrire les liens entre les interfaces locales et les interfaces distantes. L'utilisation du RSS qui peut être envisagée au niveau du protocole OLSR est que, si le RSS est accompagné d'informations suffisantes fournies par la couche liaison comparativement au contenu des messages "Hello", ces informations seront utilisées pour maintenir à jour l'ensemble local de liens à la place de l'échange périodique des messages "Hello".

- Utilisation de l'AACL "Notification" d'acquittement de la couche liaison : le protocole OLSR est conçu sans s'imposer ou espérer une spécification venant de la couche liaison. Toutefois, si l'information de la couche liaison décrivant la cassure d'un lien est disponible, un nœud peut l'utiliser. Si l'information de la couche liaison décrivant l'état de la connectivité avec les nœuds voisins est disponible, comme par exemple la perte de connectivité due à l'absence des acquittements de la couche liaison, cette information est utilisée en complément des messages "Hello" pour maintenir à jour la "Neighbor Information Base" et le "MPR Selector Set". Sur réception d'une notification de la couche liaison indiquant que le lien entre le nœud et l'interface du nœud voisin est coupé, les actions suivantes sont exécutées dans le cadre de la perception des liens : chaque tuple du "local link set" doit comporter en plus des champs déjà spécifiés plus haut, un champ temporisateur déclarant qu'un lien est perdu lorsqu'il devient incertain. Si le temporisateur n'est pas expiré, le lien est désigné par un type LOST_LINK particulier et il n'est pas considéré comme lien symétrique dans la mise à jour de l'"Associated Neighbor Tuple". la génération des messages Hello doit considérer ces champs. Les notifications de la couche liaison et le traitement de l'instabilité des liens peuvent coexister dans le protocole OLSR.
- Utilisation de l'AACL "Mise à disposition" du SNR : Chaque tuple du "Local Link Set" est accompagné de 3 autres champs qui sont, le champ "L_link_pending", le champ "L_link_quality" et le champ "L_LOST_LINK_time". Le lien entre un nœud et certains nœuds voisins peut devenir mauvais, juste après le passage des messages "Hello". Dans ce cas, la "neighbor information base" peut contenir un mauvais lien pendant au moins sa durée de validité. Les stratégies d'instabilité suivantes doivent être adoptées pour contrer une telle situation : pour chaque interface NI d'un nœud voisin en liaison avec l'interface I du nœud courant, un champ décrivant la qualité du lien est associé à ce tuple. La valeur de ce champ qui décrit la qualité du lien est comparée à 2 seuils fixés qui déterminent la valeur des champs ajoutés ci–dessus au "local link set". Dans l'implantation de base, une estimation de la qualité du lien doit être maintenue et stockée dans le champ "L_link_quality". Si une mesure du ratio signal/bruit d'un message reçu est disponible, par exemple comme notification de la couche liaison, il peut être utilisé après normalisation. Si le ratio signal/bruit n'est pas disponible de même qu'aucune autre information sur la qualité du lien provenant de la couche liaison, un algorithme peut être utilisé comme par exemple la moyenne mobile du taux de transmission avec succès. La perte d'un paquet OLSR est détectée par analyse des numéros de séquence des paquets perdus par interface et la période de long silence d'un nœud. Une longue période de silence peut être détectée si aucun paquet OLSR n'est reçu durant l'intervalle d'émission de messages "Hello". Une perte de paquet OLSR peut être détectée.
- L'extension de QoS rajoutée à OLSR à travers la version QOLSR [AGH06] permet de supporter divers critères de sélection de route, sans ajout de trafic de contrôle en dehors des messages déjà utilisés. Les MPRs à qualité de service (QMPRS) sont choisis à partir des métriques locales de QoS associées aux liens. QOLSR permet de découvrir des chemins contenant uniquement des nœuds intermédiaires QMPRs entre une source et une destination. La prise en compte de ce routage à qualité de service du protocole est différente du recours aux services de QoS recensés dans le tableau ci–dessous.

V.4.1.3. Tableau d'interaction des protocoles

Action Atomique Cross–layer (AACL)	Protocoles					
	Application	**TCP**	**OLSR**	**IP**	**Liaison 802.11**	**Physique 802.11**
"Notification" de la gigue d'envoi des paquets		D			S	
"Notification" d'évitement de retransmission		D	D		S	
"Notification" d'acquittement		D2	D2		D1 ,S2	S1
"Notification" explicite de congestion		D		S distante		
"Notification" de la baisse significative du niveau d'énergie	D	D	D	D	D	S
"Mise à disposition" de la liste des nœuds MPR		D	S			
"Mise à disposition" de la liste MPR Selector Set		D	S			
"Mise à disposition" de la liste des nœuds voisins à 1 saut		D	S			
"Mise à disposition" de la liste des nœuds accessibles		D	S			
"Mise à disposition" de la liste noire		D	S			
"Mise à disposition" du taux de perte de paquet	U	U			S	
"Mise à disposition" du SNR (Signal to Noise Ratio)	U	U	U		U	S
"Mise à disposition" du RSS (Received Signal Strength)	U	U	U		U	S
"Mise à disposition" du taux d'erreur bit BER (Bit Rate Error)	U	U			U	S
"Mise à disposition" du niveau d'énergie	U	U	U	U	U	S
Service RSVP "Activable" de contrainte de délai	X			X	X	
Service VMAC "Activable"	U/X				U/X	S
Service IntServ "Activable"	U/X			S/X		
Service DiffServ "Activable"	U/X			S/X		
Service FEC (Forward Error Correction) "Activable"		U			S	
Service ARQ (Automatic repeat request) "Activable"		U			S	

Table V.2. Tableau des interactions des protocoles avec routage OLSR.

V.4.2. Cas du protocole DSDV

V.4.2.1. Choix de la pile de protocoles

Les modifications apportées à la pile de protocoles utilisée au chapitre II concernent l'utilisation du protocole DSDV à la place du protocole DSR pour assurer le routage d'information. Les protocoles TCP, IP et 802.11 ne sont pas modifiés.

V.4.2.2. Recensement des AACL

Le protocole DSDV met en œuvre divers mécanismes pour son fonctionnement. La description de ces mécanismes permet de procéder au recensement des AACL qu'il génère. La liste des AACL ci–dessous peut être établie à partir de ces mécanismes et permet de définir la contribution du protocole DSDV aux modèles conceptuels cross–layer.

- L'AACL de "Mise à disposition" de la table de routage : en utilisant le protocole DSDV pour assurer le routage d'information, chaque nœud du réseau ad–hoc crée et maintient une table de routage. Cette table contient toutes les destinations disponibles associées à leur métrique, ainsi que le prochain saut menant à cette destination et un numéro de séquence généré par prochain saut par le nœud qui détient la table. Les paquets sont transmis dans le réseau à partir des informations de cette table. L'aspect dynamique du réseau est pris en compte. La table est mise à jour à partir des messages périodiques ou lorsqu'une information nouvelle et significative est disponible pour maintenir sa cohérence. Les paquets de mise à jour sont diffusés par chaque nœud par broadcasting ou multicasting avec une mise à jour de chaque paquet par incrémentation de sa métrique à chaque saut pendant sa phase de propagation à travers le réseau. Ce processus de propagation occasionne la conservation pendant un temps du paquet mis à jour, dans l'attente d'une meilleure route, avant qu'il ne soit pris en compte pour la mise à jour de la table de routage locale et pour sa rediffusion. Lorsque le nœud reçoit plusieurs paquets de mise à jour dans la période d'attente, le paquet ayant le numéro de séquence le plus récent est toujours préféré par la décision de diffusion. Pour un même numéro de séquence, le paquet ayant la métrique la plus faible est préféré et est utilisé pour la mise à jour et la propagation à travers le réseau. L'avertissement des routes qui doivent changer doit attendre un temps fixé pour permettre de trouver une meilleure route. Deux types de paquets de mise à jour sont utilisés, le premier appelé "full dump" contient toutes les informations de routage disponibles, et le second appelé "incremental" contient uniquement les informations ayant changé depuis le dernier "full dump". Chaque nœud du réseau ad–hoc doit périodiquement diffuser l'intégralité de sa table de routage (full dump) à ses voisins en utilisant de multiples NPDU (Network Protocol Data Units). La fréquence de la transmission peut être faible lorsque le nœud se déplace. La mise à jour complémentaire d'informations de routage se fait avec une seule NPDU. Lorsque le nombre de changements significatifs augmente avec la variation de la topologie du réseau, et que la taille du paquet "incremental" approche la taille maximale d'un NPDU, un full dump est programmé pour réduire la taille du prochain paquet "incremental".
- L'AACL de "Mise à disposition" de la liste des nœuds inaccessibles : l'adaptation au changement de la topologie se fait par détection de la cassure des liens lorsque le nœud mobile se déplace ou est mis hors service. Le lien hors service est détecté par le matériel de communication ou par absence de diffusion provenant de ce nœud pendant une période donnée. La métrique d'un lien hors service est infinie, cette valeur est donnée à toutes les routes passant par le lien en question. Le nœud qui détecte ce changement diffuse immédiatement un paquet de mise à jour et divulgue les routes modifiées avec la métrique infinie et un numéro de séquence récent. Le lien menant à un nœud ayant été perdu est rétabli lorsque le nœud diffuse un message de mise à jour avec un numéro de séquence égal

au dernier ou est plus récent et une métrique finie. Ce message est disséminé à travers le réseau. L'entrée ayant une métrique finie remplace dans tous les cas celle ayant une métrique infinie dans la table de routage.

- L'AACL de "Mise à disposition" de la liste des nœuds unidirectionnels : le protocole DSDV suppose que les liens sans fil du réseau ad–hoc sont bidirectionnels. Mais les connexions sans fil peuvent être asymétriques. La présence des liens unidirectionnels engendre 2 problèmes au niveau de DSDV. Le premier problème a trait à la connaissance asymétrique des liens, en ce sens que les nœuds destinataires connaissent l'existence des nœuds sources. Mais, les nœuds sources ne peuvent pas supposer l'existence à priori des nœuds destinations. Le second problème a trait à l'inaccessibilité des nœuds destinations, car en utilisant le protocole DSDV, chaque nœud de destination peut initier la mise à jour du chemin qui le mène au nœud source, mais sur des liens unidirectionnels, il se peut que le nœud de destination n'ait aucun moyen de diffuser son existence au nœud émetteur. Une des solutions proposées consiste à ce que chaque nœud maintienne suffisamment d'informations pour distinguer les liens bidirectionnels et unidirectionnels avec comme inconvénient majeur d'augmenter l'overhead de communication et de stockage.

Le protocole DSDV procède à la mesure de la complexité en terme de temps. La mesure de cette complexité est donnée par le diamètre du réseau. De même, la complexité en terme de communication est également mesurée par DSDV. Elle est due à l'ajout et à la perte de chemin et est donnée par le nombre de nœuds du réseau. A priori, ces deux paramètres ne font pas l'objet d'une exploitation par l'une des couches de la pile de protocoles. Pour compléter l'étape de recensement, nous définissons ci–dessous les informations cross–layer fournies par les autres couches et utilisées par le protocole DSDV.

- Utilisation des services RSVP, Intserv et DiffServ "Activable" : le routage à QoS peut être utilisé avec DSDV pour prendre en compte le trafic temps réel dans un réseau à sauts multiples. De façon générale, les besoins de routage à QoS dans un réseau ad–hoc qui tient compte d'un trafic temps réel sont énoncés dans les quatre points suivants. Le premier point concerne la réservation de la bande passante. A cet effet, le réseau ad–hoc doit allouer la bande passante a "l'appel" pour le démarrage du trafic. Le second point concerne le routage à QoS proprement dit. Les nœuds mobiles doivent connaître le chemin à délai minimal menant à la destination et nécessite de connaître la bande passante disponible sur cette route. A l'appel pour le démarrage du trafic, la bande passante doit être disponible et réservée, sinon la requête d'appel sera rejetée. De ce fait, les algorithmes traditionnels de routage à vecteur de distance ne sont pas adaptés. Ce routage à QoS est requis pour une gestion efficace des ressources. Le troisième point du routage à QoS concerne le contrôle de congestion. Bien que l'utilisation du routage à QoS gère le débit, la congestion du réseau due à la mobilité et au modèle dynamique du trafic doit être contrôlée en appliquant la perte sélective de paquets et le contrôle du débit d'entrée, … La mobilité constitue le quatrième point de la problématique du routage à QoS. L'association de la mobilité et de l'allocation et de la maintenance du débit est un facteur critique pour les réseaux ad–hoc, surtout lorsqu'il est interconnecté à un réseau filaire. La garantie de la bande passante est le besoin le plus critique des applications temps réel. Le protocole DSDV peut être associé aux services RSVP, IntServ et Diffserv.
- Utilisation de l'AACL "Notification" d'acquittement de la couche liaison : le fonctionnement du protocole DSDV révèle l'utilisation implicite de la notification d'acquittement, notamment pour la déduction des liens unidirectionnels.

V.4.2.3. Tableau d'interaction des protocoles

Action Atomique Cross–layer (AACL)	Protocoles					
	Application	**TCP**	**DSDV**	**IP**	**Liaison 802.11**	**Physique 802.11**
"Notification" de la gigue d'envoi des paquets		D			S	
"Notification" d'évitement de retransmission		D	D		S	
"Notification" d'acquittement		D2	U		D1 ,S2	S1
"Notification" explicite de congestion		D		S distante		
"Notification" de la baisse significative du niveau d'énergie	D	D	D	D	D	S
"Mise à disposition" de la table de routage		D	S			
"Mise à disposition" de la liste des nœuds inaccessibles		D	S			
"Mise à disposition" de la liste des nœuds unidirectionnels		D	S			
"Mise à disposition" du taux de perte de paquet	U	U			S	
"Mise à disposition" du SNR (Signal to Noise Ratio)	U	U			U	S
"Mise à disposition" du RSS (Received Signal Strength)	U	U			U	S
"Mise à disposition" du taux d'erreur bit BER (Bit Rate Error)	U	U			U	S
"Mise à disposition" du niveau d'énergie	U	U	U	U	U	S
Service RSVP "Activable" de contrainte de délai	X		U	X	X	
Service VMAC "Activable"	U/X				U/X	S
Service IntServ "Activable"	U/X		U	S/X		
Service DiffServ "Activable"	U/X		U	S/X		
Service FEC (Forward Error Correction) "Activable"		U			S	
Service ARQ (Automatic repeat request) "Activable"		U			S	

Table V.3. Tableau des interactions des protocoles avec routage DSDV.

V.4.3. Cas du protocole AODV

V.4.2.1. Choix de la pile de protocoles

A cette troisième phase de déduction de l'apport des protocoles de routage aux modèles conceptuels cross–layer, les modifications apportées à la pile de protocoles utilisée au chapitre II se rapportent au remplacement du protocole DSR par le protocole AODV pour assurer le routage d'information. Les autres protocoles de la pile, TCP, IP et 802.11 ne sont pas modifiés.

V.4.2.2. Recensement des AACL

Comme les autres protocoles de routage, le protocole AODV génère des informations utilisables par les autres couches. L'analyse de ses mécanismes fonctionnels permet d'établir sa contribution aux modèles cross–layer constituée essentiellement de la liste des AACL recensées ci–dessous.

- L'AACL de "Mise à disposition" de la table de routage : bien que de nature réactive, le protocole AODV entretien une table de routage dont la durée de vie est un paramètre fixé. Lorsqu'une table de routage expire, elle est supprimée. La création ou la mise à jour d'une route se fait sur la base du numéro de séquence attribué à chaque paquet pour éviter les boucles. Tout comme pour la table de routage elle–même, une route quelconque de cette table dispose d'une durée de vie fixée. La table de routage contient également le nombre de sauts menant à chaque destination.
- L'AACL de "Mise à disposition" de la liste des nœuds précurseurs : en utilisant le protocole AODV, chaque nœud du réseau maintient une liste de nœuds précurseurs. Ces nœuds sont des nœuds voisins vers lesquels une réponse de route a été générée ou acheminée. Les nœuds précurseurs d'un nœud donné sont informés par ce nœud lorsqu'il détecte la perte d'un saut d'une route.
- L'AACL de "Mise à disposition" de la liste noire : les réponses de route (notées RREP) sont des paquets envoyés par un nœud utilisant le protocole AODV, généralement en réponse à des requêtes de routes (notées RREQ), pour indiquer le chemin menant à une destination donnée. Les acquittements des RREP ne contiennent pas d'information sur les RREP acquittées. Ils arrivent juste après l'envoi des RREP avec le bit "A" positionné. Cette information est censée suffire pour désigner le fait que le lien est bidirectionnel. Si la transmission d'une RREP échoue, par absence des acquittements de la couche liaison ou réseau, le prochain nœud de la RREP est placé dans une liste noire. Le chemin emprunté par le paquet RREQ ayant engendré la réponse de route est donc unidirectionnel. La particularité d'un nœud figurant dans la liste noire d'un nœud donné est que toutes les RREQ venant de ce nœud de la liste noire seront ignorées par le nœud propriétaire de la liste. La durée de présence d'un nœud dans la liste noire est limitée.
- L'AACL de "Mise à disposition" de la liste des nœuds voisins : lorsqu'un nœud fait partie d'une route active, il doit offrir des informations de connectivité en diffusant des messages "Hello" locaux avec un TTL égal à 1. L'intervalle de diffusion a une durée fixée. Le nœud détermine la connectivité en écoutant les paquets diffusés par ses voisins. Lorsqu'aucun paquet provenant d'un nœud voisin n'a été reçu durant une période déterminée, le lien menant à ce nœud est supposé perdu. Lorsqu'un nœud reçoit un message "Hello" de la part d'un autre nœud, il doit s'assurer qu'il a une route active menant à ce nœud, ou en crée une si nécessaire. Si la route existe, sa durée de vie doit être incrémentée et doit contenir le dernier numéro de séquence. Les routes créées au moyen des messages "Hello" et non utilisées par aucune autre route active a une liste de précurseurs vide. Ces routes ne doivent pas générer de messages d'erreur de route (notés RERR) si le nœud voisin se déplace hors de portée et que sa temporisation expire.

- L'AACL de "Mise à disposition" de la liste des nœuds inaccessibles : le message d'erreur de route est un paquet initié lorsque le nœud détecte la coupure du lien menant au prochain saut d'une route active lors de la transmission des données, ou lorsque le nœud reçoit un paquet à destination d'un autre nœud pour lequel il n'a pas de routes actives, ou s'il reçoit un RERR de la part d'un nœud voisin pour une ou plusieurs route active. De façon générale, le message RERR peut être diffusé, faire l'objet d'un envoi unique à destination du seul précurseur ou de plusieurs envois uniques à destination de chacun des précurseurs. La liste des nœuds non accessibles est établie à partir de la table de routage. Un message RERR contenant les destinations pour lesquelles la liste des précurseurs n'est pas vide est envoyé aux nœuds précurseurs. La table de routage est mise à jour par les opérations de modification du numéro de séquence, de marquage de la route comme invalide ou de suppression de la route après sa durée de vie.
- L'AACL de "Notification" de la gigue d'envoi due à une requête de route : une requête de route appelée RREQ est diffusée lorsque le nœud n'a pas de route valide menant à une destination sollicitée. La RREQ est mise dans un tampon pour éviter que le même paquet acheminé par d'autres nœuds soit à nouveau traité et rediffusé. Le nombre de messages RREQ par seconde est limité. Lorsqu'aucune route n'est reçue en réponse dans un intervalle donné, le nœud diffuse un autre paquet RREQ jusqu'à un maximum de tentatives fixé par la valeur maximale du TTL. A chaque tentative, le champ TTL du paquet IP est modifié pour permettre de contrôler la distance de diffusion du paquet RREQ. Les paquets de données en attente sont mis en zone tampon en respectant le mode FIFO. Si aucune réponse de route n'est reçue durant le nombre maximal de tentatives TTL max, les données en attente pour cette destination sont supprimées et le message "destination non accessible" est envoyé à l'application. Le temps d'attente entre deux tentatives d'envoi du RREQ est calculé en utilisant un back–off exponentiel binaire : le nouveau temps d'attente a une valeur égale au double de sa valeur précédente, pour éviter de congestionner le réseau. Le contrôle de la diffusion des messages RREQ se fait par utilisation de la technique d'élargissement du cercle de recherche. Le TTL du paquet IP prend une valeur initiale et un temporisateur est démarré. La valeur initiale du TTL peut être fixée égale au nombre de sauts connus pour cette destination dans la table de routage ajoutée à une autre valeur constante fixée. A chaque expiration du temporisateur, le TTL est incrémenté d'une valeur constante fixée, jusqu'à un seuil de TTL à partir duquel une valeur constante est utilisée pour le TTL. Pour un nœud intermédiaire, le traitement et l'acheminement des RREQ passe par des étapes de vérification. Si un RREQ du même nœud a été reçu dans un intervalle donné, alors le paquet est ignoré. Sinon, le compteur du nombre de sauts du RREQ est incrémenté. La route inverse est créée ou mise à jour dans le cache pour l'acheminement éventuel d'un RREP reçu pour le RREQ courant. Si le nœud intermédiaire n'a pas généré de RREP et que le TTL du paquet IP est supérieur à 1, le nœud diffuse le paquet après avoir décrémenté le TTL. Si le RREQ contient le drapeau "G" positionné pour une réponse gratuite de route, un nœud intermédiaire peut renvoyer la RREP au nœud émetteur. Le nœud intermédiaire doit de ce fait renvoyer une route unicast au nœud destination. En réponse à un paquet RREQ, un nœud émet une RREP s'il est le nœud destination ou s'il dispose d'une route valide menant à la destination, à condition que le paquet RREQ ne dispose pas de l'option "D" activée (pour "destination only"). Lorsque le RREP est renvoyé à l'initiateur du RREQ, le champ compteur de saut est incrémenté à chaque saut. Ainsi, à destination, ce champ représente la distance en nombre de sauts de la destination à l'émetteur. Le nœud intermédiaire qui émet une RREP met à jour sa liste de précurseurs.
- L'AACL de "Notification" de la gigue d'envoi due à une erreur de route : les messages RERR d'erreur de route, d'expiration ou de suppression de route peuvent être utilisés lorsqu'un nœud d'une route ne peut pas être joint lors de la transmission d'un paquet.

En dehors des informations qu'il fournit à travers les AACL recensées ci–dessus, le protocole AODV utilise les informations fournies par les interactions des autres couches. La liste des interactions utilisées par AODV est donnée dans les points suivants.

- Utilisation de l'AACL de "Notification" d'acquittement de la couche liaison : tout nœud qui achemine le trafic doit garder les traces de ses prochains sauts et précurseurs actifs pendant une durée déterminée. Il doit également garder les nœuds ayant envoyé les messages "Hello". Pour cela, le nœud peut utiliser les mécanismes disponibles au niveau des couches liaison et réseau. L'absence d'acquittements de la couche liaison du 802.11 ou l'impossibilité d'obtenir un CTS après un RTS sont des événements qui peuvent être utilisés. La disponibilité des notifications de la couche 2 permet d'utiliser les acquittements passifs. Dans ce cas, le prochain nœud est supposé acheminer le paquet en écoutant le canal pour déterminer la tentative d'envoi du prochain nœud. Si la transmission n'est pas détectée pendant un temps donné, ou que le prochain saut est la destination (qui ne doit pas acheminer le paquet), l'une des méthodes suivantes est utilisée : réception d'un quelconque paquet du prochain saut (y compris "Hello"), un RREQ unicast pour le prochain nœud demandant un chemin qui mène à lui–même, un message unicast de requête d'Echo ICMP à destination du prochain saut. Si aucune méthode ne donne de résultat, le lien est supposé perdu et des actions correctives sont prises comme spécifié dans l'utilisation de l'AACL suivante.
- Utilisation de l'AACL de "Notification" pour la récupération d'un paquet de la couche liaison : lorsqu'une coupure du lien survient dans une route active, le nœud se trouvant en amont de la coupure choisit de réparer localement la coupure si la destination n'est pas éloignée d'un nombre de sauts fixé. L'AACL courante permet à la couche liaison de solliciter la réparation locale d'une route par la couche réseau. Dans ce cas, le nœud incrémente le numéro de séquence de la destination et diffuse un RREQ avec un TTL calculé à cet effet. Les données du paquet en instance sont mises en zone tampon. un RERR est renvoyé pour cette destination lorsqu'un RREP n'est pas reçu à la fin de la période d'attente. De même, un RERR est émis avec le bit "N" lorsqu'un paquet RREP est reçu mais contient un nombre de sauts supérieur à celui du paquet initial. Ces événements déclenchent la mise à jour de la table de routage. Le nœud émetteur doit réinitier une découverte de route s'il est voisin de l'émetteur du RERR, dans le cas contraire, il doit simplement retransmettre son paquet.

V.4.2.3. Tableau d'interaction des protocoles

Action Atomique Cross–layer (AACL)	Protocoles					
	Application	**TCP**	**AODV**	**IP**	**Liaison 802.11**	**Physique 802.11**
"Notification" de la gigue d'envoi des paquets		D			S	
"Notification" d'évitement de retransmission		D	D		S	
"Notification" d'acquittement		D3	D2, S3		D1 ,S2	S1
"Notification" explicite de congestion		D		S distante		
"Notification" de la baisse significative du niveau d'énergie	D	D	D	D	D	S
"Notification" pour la récupération d'un paquet			D		S	
"Notification" de la gigue d'envoi due à la défaillance d'une route		D	S			
"Notification" de la gigue d'envoi due au changement de route		D	S			
"Mise à disposition" de la table de routage		D	S			
"Mise à disposition" de la liste des nœuds précurseurs		D	S			
"Mise à disposition" de la liste noire		D	S			
"Mise à disposition" de la liste des nœuds voisins		D	S			
"Mise à disposition" de la liste des nœuds inaccessibles		D	S			
"Mise à disposition" du taux de perte de paquet	U	U			S	
"Mise à disposition" du SNR (Signal to Noise Ratio)	U	U			U	S
"Mise à disposition" du RSS (Received Signal Strength)	U	U			U	S
"Mise à disposition" du taux d'erreur bit BER (Bit Rate Error)	U	U			U	S
"Mise à disposition" du niveau d'énergie	U	U	U	U	U	S
Service RSVP "Activable" de contrainte de délai	X			X	X	
Service VMAC "Activable"	U/X				U/X	S
Service IntServ "Activable"	U/X			S/X		
Service DiffServ "Activable"	U/X			S/X		
Service FEC (Forward Error Correction) "Activable"		U			S	
Service ARQ (Automatic repeat request) "Activable"		U			S	

Table V.4. Tableau des interactions des protocoles avec routage AODV.

V.4.4. Synthèse des AACL recensées

L'application des trois premières étapes de la méthode RCL aux piles de protocoles qui se différencient par l'utilisation de l'un des quatre protocoles de routage sélectionnés, a permis de faire ressortir les contributions de ces protocoles en terme d'interactions cross–layer. Ces quatre protocoles génèrent des AACL de mise à disposition pour rendre les données liées à leurs structures de routage accessibles aux autres couches. Ces structures de routage sont différentes d'un protocole à un autre et dépendent de la philosophie fonctionnelle de chaque protocole. L'étude comparative à mener doit en faire une synthèse et faire ressortir les données réellement utilisées par les autres couches de la pile de protocoles.

En dehors des AACL de mise à disposition, les protocoles de routage utilisent de diverses façons les données des interactions cross–layer fournies par les autres couches. Le protocole OLSR par exemple utilise le ratio signal à bruit provenant de la couche liaison pour déduire la qualité d'un lien du réseau. Le protocole AODV n'en fait aucun usage explicite. Dans un deuxième exemple, les protocoles AODV et OLSR utilisent les notifications d'acquittement de la couche liaison tandis que DSDV n'en fait qu'un usage implicite. La différence de fonctionnement d'un protocole à un autre explique cette variation de l'usage des données fournies par les autres couches de la pile. Certains protocoles mettent en œuvre des mécanismes internes qui leur sont propres, tel le cas du protocole DSR qui prévoit l'implantation des acquittements explicites, malgré l'utilisation possible des acquittements de la couche liaison, pour s'assurer une indépendance de fonctionnement. La consommation faite par les protocoles de routage des données des AACL provenant des autres couches n'a pas d'impact dans le modèle global, en ce sens que l'absence d'une telle consommation ne modifie pas le comportement du système cross–layer. En revanche, les données fournies par ces protocoles et utilisées par d'autres couches ont une influence notoire sur le fonctionnement du système cross–layer, influence qui peut s'avérer négative lorsque ces données sont absentes. C'est pourquoi en plus d'expliciter la nature des données fournies par les protocoles de routage et qui sont réellement utilisées par les autres couches, il est important de s'assurer de leur disponibilité quelque soit le protocole de routage utilisé.

Parmi les protocoles sélectionnés, ceux de la famille de routage ré–actif mettent en œuvre des mécanismes internes qui favorisent la mise en œuvre des AACL de notification. C'est le cas des notifications de gigue provenant des messages échangés par ces protocoles lors de la transmission d'un paquet. Dans le cas du routage pro–actif, ce rôle est assuré par des mécanismes complémentaires provenant du protocole IP, à savoir l'utilisation des messages retour ICMP.

V.5. La table LAST (Link Access State Table) du sous–système environnement

L'analyse des données fournies par les protocoles de routage reflète la mise à jour des informations de routage sous diverses formes. Elle reflète également l'envoi d'information à destination des couches supérieures pour des événements qui prolongent les délais de délivrance des paquets en cours de transmission. La couche transport est la principale couche consommatrice de ces interactions. Les données ne sont pas les mêmes d'un protocole à un autre. Toutes ces données ne sont pas consommées dans leur forme de routage par la couche transport. Par exemple, il peut être intéressant de savoir au niveau transport qu'un nœud est voisin direct d'un nœud donné pour permettre d'utiliser certaines informations de la couche liaison comme les acquittements par exemple ou les autres informations d'état du lien.

L'utilisation du protocole OLSR donne les AACL relatives à la liste des nœuds voisins à 1 saut, à la liste des nœuds MPR et à la liste MPR Selector Set. Ces trois listes ramènent au même concept de proximité d'un nœud par rapport à un autre tout comme la liste des nœuds précurseurs d'AODV ou la liste CS du protocole CONSET. La liste noire d'OLSR, de DSR, d'AODV et la liste des nœuds inaccessibles d'AODV et de DSDV reflètent les nœuds du réseau qui ne peuvent être joints pour une communication. La liste des nœuds accessibles donnée par le protocole OLSR peut être identifiée à la table de routage fournie par les protocoles DSR, DSDV et AODV.

Les structures de routage fournies par les protocoles contiennent des données dans des formats différents et dont toutes ne sont pas utilisées par la couche transport. Par exemple, les informations sur l'interface ou sur le nœud qui a annoncé une route ne font pas l'objet d'un usage particulier. C'est pourquoi il est possible de restreindre les informations fournies par les AACL des protocoles de routage afin de distinguer trois catégories d'informations utiles, qui sont :

- la liste des nœuds voisins à 1 saut,
- la liste des nœuds accessibles à plus d'un saut,
- la liste des nœuds inaccessibles.

Le premier effort de standardisation à réaliser pour rendre transparent le changement du protocole de routage sera de générer ces trois tables au niveau du sous–système environnement, à partir des informations cross–layer fournies par n'importe quel protocole de routage.

Pour rendre le routage transparent, la couche transport ne doit pas considérer la nature pro–active ou ré–active des protocoles de routage du réseau ad–hoc. Par exemple, le protocole DSR de nature ré–active peut détecter la gigue d'envoi due à un changement de route grâce à son mécanisme de récupération. Le protocole OLSR, de nature pro–active, permet de récupérer un paquet lorsqu'un nœud de son chemin est hors d'usage mais ne fait pas de l'avertissement de l'émetteur une obligation. Tous les protocoles détectent la défaillance d'une route du fait des messages d'erreur de route qu'ils véhiculent, y compris ceux qui ont recours au mécanisme complémentaire d'ICMP. Dans le cas d'une erreur de route pour laquelle aucune route alternative n'existe dans la table (par épuisement des routes possibles ou implantation de la politique des tables à route unique), les protocoles ré–actifs initient une découverte ou une requête de route. Les protocoles pro–actifs procèdent au marquage de la route comme étant défectueuse ou la supprimeront de la table de routage en fonction de la politique de gestion implantée. Ils pourront également utiliser une route alternative lorsque la multiplicité de routes est autorisée dans les tables de routage. Cependant, la mise à jour d'une nouvelle route lorsque le protocole pro–actif n'en dispose pas attendra généralement la prochaine période de rafraîchissement.

La problématique de la transmission des paquets peut être perçue sous divers angles en utilisant un protocole pro–actif ou ré–actif. Les protocoles ré–actifs peuvent être exploités pour notifier les gigues d'envoi dues à la défaillance d'une route ou à une erreur de route. L'objectif sera de permettre à la couche transport fiable de réinitialiser ses temporisateurs d'attente d'envoi pour éviter leur expiration qui serait préjudiciable au débit de données. En rappel, la prise en compte de l'évolution des temporisateurs pour éviter leur expiration a été le point clé des versions TCP Reno et NewReno. Contrairement à ce principe d'exploitation de la notification de gigue propre aux protocoles ré–actifs, les protocoles pro–actifs n'implantent pas de mécanismes qui donnent les raisons explicites d'une erreur de route qui survient. Ils peuvent de ce fait engendrer des expirations des temporisateurs ou des échecs de transmission, surtout lorsque les gigues avoisinent la fin de la durée de vie du paquet au niveau de la couche réseau. Dans un tel cas, le temps restant peut ne pas permettre au protocole pro–actif de régler la transmission du paquet avant la fin de sa durée de vie, tandis qu'un protocole ré–actif peut

recevoir une notification de changement de route et prendre des mesures adéquates (réinitialiser ses temporisateurs, informer la couche transport) grâce à des mécanismes cross–layer. Du point de vue de la transmission des données par les protocoles de routage, trois états méritent d'être connus par la couche transport. Ces trois états sont obtenus grâce aux messages reçus par le protocole de routage :

- paquet en cours de transmission : cet état traduit le fait que le nœud n'a pas reçu de message d'erreur ou de défaillance de route lors de la transmission d'un paquet. Ce paquet est supposé suivre son chemin à travers le réseau.
- paquet en attente : lorsque le paquet attend l'exécution de mécanismes tels que la découverte de route ou l'attente de rafraîchissement des structures de routage.
- transmission du paquet abandonnée.

Pour prendre en compte ces trois états lors de la transmission d'un paquet, nous proposons d'utiliser la table LAST au niveau du sous–système environnement. Cette table aura pour fonction de servir de miroir de transmission des paquets. Elle constitue le mécanisme cross–layer standard qui permet de livrer à la couche transport les informations dont elle a besoin indépendamment de la nature pro–active ou réactive de chaque protocole de routage.

Si un protocole ré–actif est utilisé, l'émission du paquet se traduit par l'état "paquet en cours de transmission" qui lui sera associé. Le paquet reste dans cet état lorsque le protocole de routage ré–actif reçoit les informations qui lui permettent de générer les messages de notification de la gigue d'envoi. Le paquet passe à l'état d'attente lorsqu'un message d'erreur de route est reçu ou lorsque le nœud juge nécessaire de relancer une découverte de route. L'abandon signifie que le paquet a expiré sa durée de vie au niveau du protocole de routage. Dans le cas des protocoles pro–actifs, la mobilité des nœuds va engendrer un passage plus fréquent de l'état "paquet en cours de transmission" à l'état "paquet en attente" puisque les messages de notification de gigue ne sont pas implantés. Ce changement d'état sera opéré pour chaque réception d'un message d'erreur de route ou lorsque le protocole juge nécessaire d'attendre le rafraîchissement des structures de routage. La mesure du gain apporté par l'exploitation des messages de notification dépend fortement de l'intégration du critère de mobilité.

Les trois états qui caractérisent la table LAST peuvent être utilisés par la couche transport pour développer l'un des deux comportements mis en évidence, à savoir, la temporisation traditionnelle ou la temporisation persistante. Le comportement persistant peut être observé par la couche transport fiable lorsque des messages de changement de route lui parviennent, du fait de la mobilité des nœuds. De ce fait, le protocole transport fiable attendra le prochain message explicite de délivrance du paquet sans faire évoluer son temporisateur vers son expiration. La non délivrance peut être constatée par divers mécanismes dont par exemple le message explicite du protocole de routage. L'opportunité d'une relance ultérieure provenant du protocole de transport mérite d'être étudiée conformément au principe de la temporisation persistante. Les notifications de changement de route ont l'avantage d'éviter de conduire les temporisateurs à leur expiration.

V.6. Modèle cross–layer de temporisation unifiée

La liaison entre la couche transport et les protocoles de routage est assurée dans les deux sens ascendant et descendant. Le processus d'émission de paquet constitue la liaison descendante entre les deux couches. Dans le sens ascendant, le protocole de routage envoie un message d'erreur au protocole de la couche transport lorsqu'il n'arrive pas à acheminer le paquet.

A la réception du paquet en cours de transmission, les protocoles de routage exécutent diverses primitives. L'analyse de ces primitives permet de déceler les mécanismes cross–layer qui peuvent être proposés pour optimiser la transmission. C'est pourquoi nous considérons le comportement de trois protocoles de routage : le protocole DSR, le protocole AODV et le protocole OLSR.

V.6.1. Primitive de transmission de paquet du protocole DSR

Lorsqu'un nœud émetteur envoie un paquet à destination d'un autre nœud, le protocole DSR insère dans le paquet la route source menant à la destination. Cette route source provient du cache de route contenant les routes précédemment apprises. S'il n'y a pas de route connue pour cette destination, le processus de découverte de route est dynamiquement activé pour permettre au protocole d'obtenir la route qui fait défaut.

La découverte de route se fait en envoyant un paquet de requête de route à travers le réseau, ce paquet pouvant être le paquet de données à transmettre auquel le protocole ajoute son entête ou une copie du paquet sans les données. Dans ce dernier cas, le paquet original est stocké dans le tampon d'attente d'envoi appelé "Send Buffer". Ce tampon contient pour chaque paquet en attente d'envoi, le temps d'insertion du paquet. Le paquet est supprimé du tampon d'attente d'envoi après une période délimitée par un temporisateur. Pour éviter un débordement possible de ce tampon, une politique FIFO de sortie peut être appliquée ou toute autre stratégie. Ces stratégies auront toutes pour conséquence de supprimer les paquets les plus vieux en attente d'envoi même si le temporisateur de conservation n'a pas expiré.

Tant que le temporisateur d'attente d'envoi du paquet n'a pas expiré, le nœud doit initier occasionnellement de nouvelles découvertes de route pour chaque destination. Le nœud doit limiter la fréquence de lancement de ces découvertes de route puisqu'il est possible que le nœud soit inaccessible pendant une période assez longue (portée limitée de la transmission sans fil, mobilité des nœuds, ...), également pour éviter qu'un nombre assez grand de paquets de découverte de route improductifs soient injectés dans le réseau. Pour réduire l'overhead, la limitation de la fréquence des découvertes de route d'une même destination se fera à travers l'algorithme de back–off exponentiel. La valeur du temporisateur d'attente sera doublée entre deux tentatives successives. Si les couches supérieures tentent d'envoyer des données supplémentaires pour cette même destination (cas du SCTP où les acquittements peuvent être groupés, ou de UDP), les paquets doivent être placés dans le tampon d'attente d'envoi jusqu'à réception d'une réponse de route. Mais en aucun cas le nœud ne doit initier de nouvelles découvertes de route sans attendre l'expiration du temporisateur qui détermine l'intervalle minimal entre deux tentatives. Cette limitation de la fréquence maximale des découvertes de route pour une même cible est la même que celle du mécanisme requis dans le réseau Internet pour limiter la fréquence à laquelle les requêtes ARP sont envoyées pour un même nœud cible.

L'envoi par DSR d'un paquet provenant de la couche transport est soumis à une durée maximale dans le tampon d'attente d'envoi, mais également à une durée minimale dans le tampon en cas de saturation. Le paquet est également soumis à des intervalles de tentative de découverte de route qui sont donnés par l'expiration d'un temporisateur soumis au back–off exponentiel binaire du fait qu'il acquiert une valeur double de sa valeur précédente à chaque expiration.

V.6.2. Primitive de transmission de paquet du protocole AODV

Lorsque le protocole AODV est utilisé pour le routage d'informations, l'émission d'un paquet par un nœud du réseau donne lieu à une diffusion du message RREQ de requête de route lorsque le nœud ne dispose pas de route valide menant à la destination recherchée. Il peut s'agir d'une destination pas encore connue du nœud source ou d'une route ayant expirée de la table de routage. La diffusion du message RREQ est précédée de la mise en cache de son numéro de séquence et de l'adresse du nœud émetteur pendant un temps déterminé, pour éviter de retraiter le même message acheminé par les autres nœuds. Un nœud donné ne doit pas générer plus d'un nombre fixé de messages RREQ par seconde. Après la diffusion du message RREQ, le nœud se met en attente des messages RREP de réponse de route pendant un intervalle de temps fixé en milli–secondes. L'absence de réponse de route durant cet intervalle engendre l'envoi d'un autre RREQ, et ainsi de suite, jusqu'à un nombre maximum de tentatives fixé. A chaque envoi de paquet RREQ, la valeur du champ TTL d'IP est incrémentée pour permettre de couvrir de façon progressive un horizon le plus large possible. La valeur du champ TTL augmente progressivement jusqu'à un maximum fixé. Les paquets de données en attente pour cette route sont mis dans un tampon géré en mode FIFO. A l'expiration sans résultat du nombre maximal de tentatives, tous les messages en attente de la route recherchée sont supprimés et le message "Destination inaccessible" est envoyé aux couches supérieures. Pour réduire la congestion du réseau du fait des tentatives répétées de découverte de route pour une même destination, le nœud utilise le back–off exponentiel binaire.

En résumé, l'envoi des paquets d'un nœud qui utilise le protocole AODV est soumis à la contrainte d'un nombre fixé de requêtes de route par seconde. De même, cet envoi est soumis au respect d'un intervalle d'attente de réponse de routes, ainsi qu'à un nombre maximal de tentatives et à un accroissement exponentiel binaire du temps d'attente entre deux émissions du message RREQ.

V.6.3. Mécanisme de transmission de paquet du protocole OLSR

OLSR utilise un routage saut par saut dans lequel chaque nœud utilise l'information locale pour router les paquets. Le protocole OLSR hérite de la stabilité de l'algorithme de routage à état de liens. Dans un cas de fonctionnement normal, les routes déterminées par le protocole OLSR sont disponibles à l'avance, et utilisables lorsque les nœuds en ont besoin. Le protocole OLSR maintient des routes menant à toutes les destinations du réseau et ne procède pas lui même à l'acheminement des paquets. Il maintient une table de routage utilisée par IP, ICMP et d'autres protocoles pour assurer l'acheminement des paquets [RFC1812]. Du fait de sa nature pro–active, le protocole OLSR gère des flux relatifs à son trafic de contrôle. La mise à jour de la table de routage est faite au moyen de ces messages périodiques de contrôle.

La temporisation d'envoi d'un paquet n'est pas utilisée par la couche réseau, contrairement aux cas précédents où les protocoles de routage ré–actifs ont été utilisés. Le protocole OLSR dispose à l'avance des routes menant à toutes les destinations du réseau avec une spécification complémentaire pour les destinations inaccessibles. Lorsqu'un paquet est émis, il est envoyé au prochain saut de la route spécifié par la table de routage et aucune autre action n'est entreprise. La gestion des tentatives infructueuses de transmission se fait au niveau des couches liaison et physique comme dans n'importe quel cas d'utilisation du 802.11. Lorsque le nombre maximal de tentatives est atteint au niveau du nœud émetteur, la couche liaison renvoie le message d'erreur de transmission à destination de la couche transport. Pour les nœuds intermédiaires chargés d'acheminer les paquets, l'échec des tentatives de transmission au niveau des couches liaison et physique engendre le renvoi des messages ICMP

spécifiant la coupure de la route. Pour gérer un tel événement, le protocole OLSR procède à la mise à jour de son "Local Link Set" en marquant le lien comme étant invalide.

V.6.4. Problématique de la redondance de la temporisation

V.6.4.1. Spécification

Les protocoles de routage pro–actif établissent les tables de routage à l'avance et n'interfèrent pas dans le processus d'envoi des paquets. Ce processus est laissé aux mécanismes traditionnels de la couche réseau constitué des protocoles de la famille IP. Ces protocoles n'utilisent pas de temporisateur d'attente à l'émission des paquets.

En revanche, lorsque les protocoles ré–actifs sont utilisés et qu'une route n'est pas disponible, une découverte de route est lancée et un temporisateur est déclenché. L'évolution de ce temporisateur décrit un back–off exponentiel généralement binaire. Ce temporisateur se trouve à une échelle de temps différente de celle du temporisateur utilisé par les protocoles de transport fiables (TCP ou SCTP). La redondance de la temporisation apparaît dans ce cas pour le même envoi avec une différence qui se situe au niveau des intervalles de relance et de la durée de vie des PDUs dans chacune des couches.

V.6.4.2. Mise à l'échelle temporelle

La conséquence de l'expiration du temporisateur d'attente d'envoi au niveau de la couche réseau se traduit par une relance du processus d'envoi jusqu'à expiration du nombre de tentatives. Cette expiration va se traduire par l'envoi d'un message d'erreur de transmission à la couche transport. Cette phase d'expiration du nombre de tentatives de la couche réseau conduit à l'expiration du temporisateur de la couche transport, puisque la relance de l'envoi par la couche transport fiable juste après la réception du message d'erreur de transmission n'est pas opportune. La couche transport fiable peut opérer une relance, uniquement lorsque son temporisateur expire et qu'elle n'a pas épuisé le nombre maximal de tentatives. A la grande différence de l'expiration du temporisateur d'envoi de la couche réseau, l'expiration du temporisateur de la couche transport fiable se traduit par des conséquences sur le débit de données (retour à la phase de démarrage lent pour le protocole TCP par exemple). Il est intéressant d'observer ce principe de redondance de la temporisation à travers les échanges cross–layer, avec comme gain escompté, celui du contrôle des retransmissions pour améliorer la consommation d'énergie et le taux de tentatives infructueuses d'émission de paquet.

La figure ci–dessous présente un cas idéal, celui où la retransmission de la couche transport a lieu après toutes les tentatives de retransmission du protocole de routage. Sachant que dans le cas où le protocole de routage ré–actif reçoit une relance de transmission d'un paquet alors que le paquet est dans son cache d'attente d'envoi, la relance est simplement ignorée et le processus d'envoi du paquet suit son cours normal.

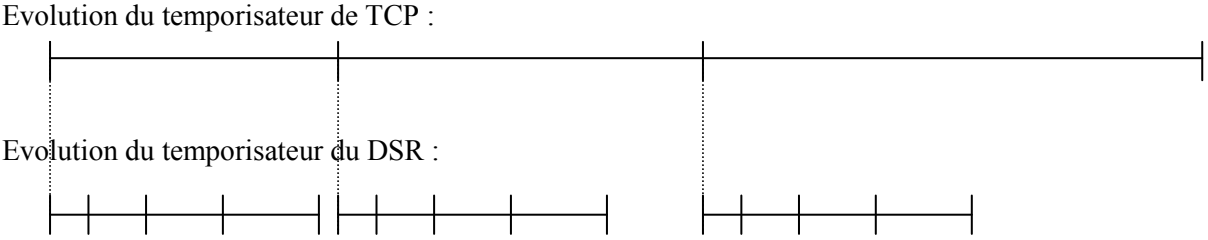


Figure V.1. Mise à l'échelle temporelle des retransmissions de TCP et du DSR

V.6.4.3. Orientation de l'étude de la redondance de la temporisation

Une application peut spécifier la durée de vie des données qu'elle envoie à la couche transport pour que si cette durée expire, les données qui n'ont pas encore été transmises soient supprimées comme dans le cas des messages de signalisation sensibles au délai. Par ailleurs, la durée de vie d'un segment au niveau de la couche transport fiable peut être déduite des temps successifs entre les expirations du temporisateur jusqu'à épuisement du nombre maximal de tentatives. En rappel et en considérant l'exemple du protocole TCP, le temps initial d'expiration du temporisateur du TCP émetteur est fixé à 3 secondes tandis que le temps de la fin d'une connexion TCP est en général fixé à 3 minutes.

Une orientation possible inscrite dans l'optique d'éviter la redondance de la temporisation, est que la durée de vie d'un segment au niveau de la couche transport soit un paramètre cross–layer commun, connu également de la couche réseau. Par analogie à l'application qui spécifie la durée de vie de ses messages au protocole de transport, ce principe de spécification de la durée de vie d'un message peut être utilisé entre la couche transport et la couche réseau lorsque le routage ré–actif est utilisé. Il s'agit pour la couche transport qui envoie un message à transmettre, de spécifier le temps global que mettra le message avant sa suppression du tampon d'envoi et la spécification de l'inaccessibilité de la destination à l'application. Lors de sa mise en œuvre, cette orientation va engendrer une modification de la politique de gestion du tampon d'attente d'envoi du protocole de routage.

Le protocole DSR par exemple intègre un champ qui indique la durée de vie d'un paquet dans son tampon d'attente d'envoi. Cependant, le paquet peut être supprimé avant l'expiration de ce temporisateur en fonction de la charge du système et une politique FIFO est appliquée pour supprimer les paquets existants qui doivent faire place aux paquets entrants.

Une première modification possible de la politique de gestion du tampon d'attente d'envoi consiste à harmoniser la durée de vie du message pour qu'elle soit la même dans les deux tampons d'attente d'envoi, à savoir celui de la couche transport et celui de la couche réseau. Pour cela, la durée de vie maximale entre les deux couches s'impose comme valeur commune. La faisabilité de cette harmonisation passe par une AACL de notification de la durée de vie du message de la couche transport au protocole de routage. La politique de suppression de message du tampon d'attente d'envoi doit être modifiée. La suppression ne doit avoir lieu qu'à l'expiration du délai maximal, y compris en faisant appel à des mécanismes complémentaires en fonction de la charge du système comme l'extension de la file d'attente d'envoi ou la génération de l'AACL de notification d'évitement de retransmission.

La deuxième modification envisageable consiste à mettre en place une file d'attente d'envoi unique dans le sous–système environnement. Chaque élément de la file unique aura des attributs multiples pour prendre en compte le fait que les messages manipulés par les protocoles n'ont pas les mêmes entêtes (segments TCP encapsulés, routage par la source pour DSR, autres champs spécifiques appartenant aux protocoles de routage). L'unicité de la file d'attente d'envoi peut aussi être assurée lorsque des mécanismes internes assurent la correspondance exacte entre un segment de la couche transport et le même paquet au niveau de la couche réseau.

Ces solutions envisageables en prélude à l'unification de la temporisation doivent garantir la transparence de l'utilisation d'un quelconque protocole de routage par rapport au fonctionnement de la couche transport fiable. Elles doivent être mise en place comme des options complémentaires cross–layer. En plus de l'évaluation des gains escomptés en termes de consommation d'énergie, de limitation du nombre de tentatives infructueuses et de non retour à

la phase de démarrage lent du fait que la congestion du réseau est réglée par un espacement efficace entre les tentatives d'émission (par analogie au comportement de l'algorithme Vegas de TCP), la mise en œuvre de la temporisation unifiée nécessite de régler le problème de sa localisation pour la rendre accessible aux deux couches (sous–système environnement par exemple). Le comportement de chaque couche doit être spécifié pour que chaque couche puisse fonctionner normalement même sans l'unification. Les deux solutions précédentes conduisent normalement à une modification du mécanisme de retransmission au niveau transport et à un étalement de la retransmission du protocole de routage sur toute la durée de vie maximale spécifiée du paquet.

V.6.4.4. Problématique des intervalles de la temporisation unifiée

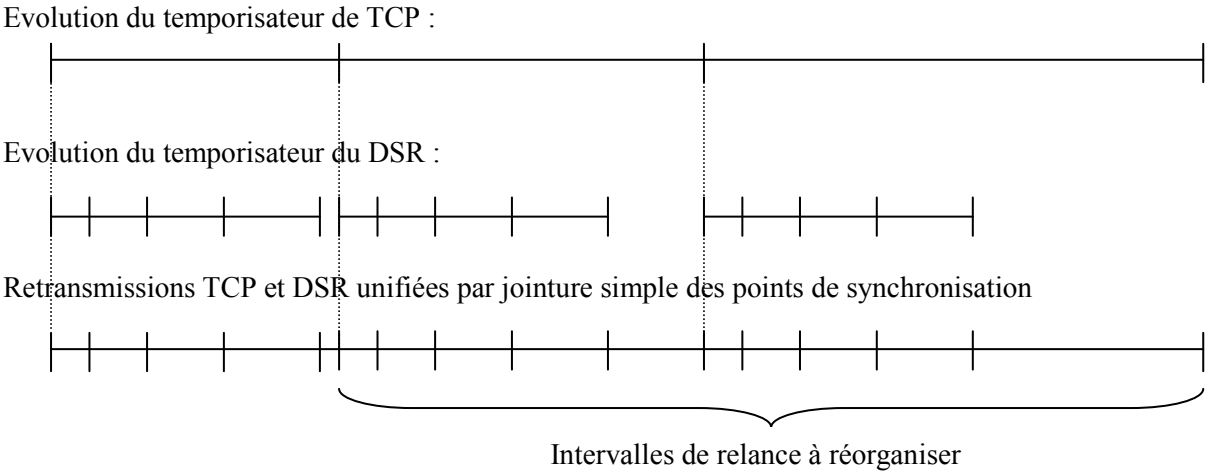


Figure V.2. Mise à l'échelle temporelle de la retransmission unifiée

A partir de l'unification simple des points de relance des deux processus de retransmission donnée par la figure ci–dessus, le problème à résoudre pour rendre efficace la temporisation unifiée est de trouver l'évolution optimale du temps d'attente entre deux tentatives de retransmissions. La meilleure fréquence de retransmission unifiée doit être trouvée en évitant d'introduire une latence d'envoi par rapport aux 2 retransmissions lorsqu'elles étaient séparées, tout en minimisant le nombre de retransmissions. La retransmission unifiée doit donc fournir un gain en économie du nombre de retransmissions pour minimiser la consommation d'énergie et le nombre de tentatives infructueuses. Pour cela, la seule évolution du temporisateur initial pour décrire un back–off exponentiel binaire ne peut se faire, car elle aura l'inconvénient majeur d'engendrer une latence due à un intervalle d'attente assez long entre deux tentatives lorsque le processus s'étale dans le temps. Le choix de la valeur des intervalles dans la partie des intervalles de relance à réorganiser doit également prendre en compte l'important paramètre de non–retour au démarrage lent du protocole transport, et éviter un encombrement du réseau. Si la congestion du réseau peut être une hypothèse écartée du fait de l'implantation de la notification explicite de congestion, de même si la temporisation persistante est adoptée pour répondre à un mauvais état du canal, l'explication de ces retransmissions peut provenir de l'absence de route ou de réponse de route suite aux requêtes de route initiées. Ces retransmissions peuvent également provenir de la modification de la route calculée par la source du paquet en cours de transmission, ou des raisons propres au mécanisme du routage d'information. La vitesse des nœuds, l'inaccessibilité des nœuds ou la variation des conditions du réseau, peuvent être les causes de ces retransmissions. Les raisons qui expliquent ces retransmissions ne sont pas maîtrisées à priori par le nœud source. Elles militent en faveur d'un comportement Vegas du protocole de transport dans lequel la fenêtre de congestion peut

augmenter d'une unité pendant le temps RTT ou rester constante ou encore diminuer au besoin, par simple interprétation de la valeur du RTT, sans retour au démarrage lent. La latence évoquée ci–dessus se rapporte non pas au décalage d'une relance par rapport à une autre équivalente lorsque les temporisations étaient indépendantes, mais se rapporte plutôt à l'étirement du temps de relance qui engendrera une perte d'activité non bénéfique.

Le recours à la modélisation mathématique peut permettre de trouver un début de solution aux bornes de relance. Le premier intervalle étendu est défini lorsque le protocole de routage a épuisé pour la première fois son nombre maximal de tentatives avec un temporisateur qui a évolué en back–off exponentiel. Le dernier intervalle de relance de ce back–off exponentiel doit être considéré comme intervalle de base pour le découpage du second intervalle étendu correspondant aux bornes de relance de TCP. Dans ce découpage destiné à faire ressortir les bornes de relance de la temporisation unifiée, l'intervalle de base peut être reconduit (ce qui fait une évolution constante) ou augmenter, en fonction de l'optimisation donnée par la distribution du nombre maximal de tentatives du protocole de routage. L'optimisation de la distribution a comme contrainte d'engendrer moins de retransmissions. Le même procédé peut être utilisé de façon récurrente pour le découpage du troisième intervalle étendu, et ainsi de suite, d'un intervalle consécutif à un autre.

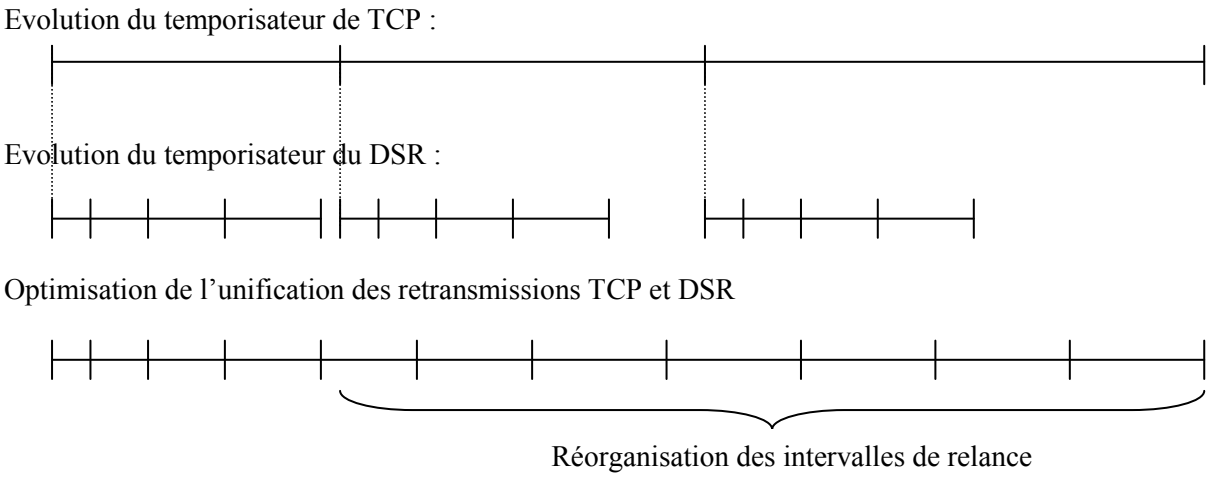


Figure V.3. Mise à l'échelle temporelle de la retransmission unifiée optimisée

V.7. Conclusion

Les protocoles fiables de la couche transport peuvent adopter un comportement persistant lors d'un mauvais état du canal. La particularité du comportement persistant est la suspension de l'activité de retransmission au niveau de la couche transport. Pour sortir de cet état, le système a besoin d'une évaluation continue de l'état du canal. L'activité ambiante peut permettre de régler ce problème d'évaluation continue, tout comme l'activité périodique des protocoles de routage pro–actif. Mais, lorsque les paramètres d'état du canal n'ont pas subi de mise à jour pendant un temps déterminé, le système doit réagir en sollicitant des mises à jour de route des paquets en cours de transmission pour permettre la poursuite de l'évaluation.

L'étude des quatre protocoles de routage choisis dans la famille des protocoles pro–actifs (OLSR et DSDV) et des protocoles ré–actifs (AODV et DSR), à travers l'application des trois premières étapes de la méthode RCL, a permis d'enrichir les modèles cross–layer avec des AACL de mise à disposition des données liées à leurs structures de routage et des AACL de notification, notamment de gigue d'envoi de paquets. Les AACL ainsi produites ont fait ressortir les points de convergence et de différence entre ces protocoles, et ont permis de définir une structure standard composées de trois tables destinées à rendre transparent le changement du protocole de routage dans le fonctionnement des modèles cross–layer. Il s'agit, pour chaque nœud considéré, de la table qui donne la liste de ses voisins à 1 saut, celle qui donne la liste des nœuds accessibles à plus d'un saut et enfin de la table relative à la liste des nœuds inaccessibles.

Egalement, de cette étude des protocoles de routage, et à partir de la problématique de la transmission des paquets telle que perçue au niveau des protocoles de transport fiable, nous avons proposé la table LAST (Link Access State Table) qui indique pour chaque paquet émis par la couche transport, l'état de l'acheminement du dit paquet. La table LAST permet d'indiquer à la couche transport fiable qu'un paquet est en cours de transmission, s'il est en attente ou si sa transmission est abandonnée. L'exploitation des données de cette table peut se faire de plusieurs façons au niveau de la couche transport, telle que par exemple le recours à la temporisation persistante lorsque des messages de changement de route dus à la mobilité des nœuds parviennent au nœud émetteur.

L'étude des primitives d'émission de paquets des protocoles de routage a permis d'observer la duplication de la temporisation d'attente d'envoi, à des échelles de temps différentes, entre la couche transport et la couche réseau lorsqu'un protocole fiable est utilisé en combinaison avec un protocole de routage ré–actif. Nous avons donc proposé dans ce chapitre, le mécanisme cross–layer de temporisation unifiée destiné à minimiser la consommation d'énergie par répétition du nombre de tentatives infructueuses. Le mécanisme de temporisation unifiée est accompagné d'un non–retour à la phase de démarrage lent pour éviter une perte de débit avec l'adoption possible d'un comportement Vegas dans lequel la valeur de la fenêtre de congestion peut prendre des valeurs stagnantes ou évolutive dans ce cas, en fonction du temps de réponse du réseau.

L'évaluation des apports de la table LAST et de la temporisation unifiée constituent la base des perspectives de la présente thèse.

Conclusion Générale

Nous consacrons ce chapitre à la synthèse des contributions apportées par la présente thèse avec à la fin, l'indication des perspectives qui se dégagent.

L'évolution actuelle des réseaux sans fil offre divers intérêts en terme de recherche. Le déploiement des modèles et mécanismes cross–layer pour les réseaux sans fil constitue un centre d'intérêt particulier dans la mesure où les propositions qui doivent être faites doivent consacrer la faisabilité ou non de la mise en place de ces modèles appuyés par un gain significatif évalué, tout comme l'abandon ou non de cette orientation de la recherche en informatique et télécommunications.

La méthode RCL de conception de modèles et mécanismes cross–layer :

Le déploiement des réseaux ad–hoc hérite des modèles et mécanismes existants dans l'environnement des réseaux câblés. L'architecture en couches est un des nombreux héritages et comporte des avantages certains qui ont occasionné la stabilité des réseaux filaires actuels. Cependant, de nombreux paramètres s'imposent pour être pris en compte dans ce processus de basculement d'un environnement à un autre, notamment, les défaillances propres à la transmission de l'information dans l'environnement sans fil. Dans ce contexte de passage de l'environnement filaire à l'environnement sans fil, tout comme celui de l'émergence des réseaux mixtes qui comportent une partie filaire et une partie sans fil interconnectées, la conception des modèles et mécanismes cross–layer est apparue pour répondre aux besoins de l'amélioration des performances des réseaux sans fil qui sont handicapés par la nature du support radio de transmission, comparativement aux réseaux câblés. Tout de même, la mise en place des réseaux filaires s'est faite de façon organisée autour de l'architecture en couches, avec un principe d'indépendance et d'ordre hiérarchique entre les couches. La conception modulaire, la définition systématique des interactions entre les composants, la poursuite des objectifs à long terme quant à l'utilisation des réseaux, etc., constituent des exemples des acquis de l'architecture. La conception cross–layer introduit quant à elle, une notion de partage d'informations et de services entre les couches, sans nécessité de respect de l'ordre architectural. L'équation est ainsi posée, la solution doit permettre de concilier les acquis de l'architecture et les gains de performances apportés par les modèles cross–layer qui s'avèrent nécessaires à l'amélioration des performances des réseaux sans fil. En réponse à cette équation, nous avons considéré la nécessité de faire évoluer la conception des modèles et mécanismes cross–layer dans un cadre standard qui aura pour objectif de favoriser l'évolution des modèles conceptuels des interactions entre les protocoles chaque fois qu'il faudra prendre en compte de nouvelles interactions. L'importance de la méthode de conception RCL que nous avons proposée se traduit également dans la mise en place de nouveaux modèles cross–layer qui font intervenir d'autres protocoles et d'autres interactions. Ainsi, la conception devient un préalable à la mise en place des systèmes cross–layer. Nous avons formalisé les démarches de cette conception à travers la méthode RCL proposée pour permettre d'obtenir des modèles conceptuels cross–layer dont l'avantage est de permettre de juger de l'impact des interactions afin de prendre en compte les acquis de l'architecture lors de la mise en œuvre mécanismes cross–layer. Les modèles d'interactions et les tableaux descriptifs des interactions sont les éléments qui résultent de l'application de la méthode RCL qui comporte sept étapes. La première étape permet de choisir la pile de protocoles pour laquelle la mise en place des mécanismes d'optimisation cross–layer est envisagée. La seconde étape se rapporte au

recensement des actions élémentaires AACL et vise à favoriser l'émergence des idées pouvant engendrer des interactions cross–layer pour améliorer la performance du système. A cette étape la pile de protocole est complétée par différents services réseaux potentiellement utilisables par les protocoles considérés. Elle permet de prendre en compte les paramètres significatifs d'une couche donnée, ainsi que les événements significatifs qui y surviennent et qui peuvent être exploités par les autres couches. La troisième étape de la méthode RCL permet de définir un tableau d'interactions des protocoles qui donne la distribution des interactions entre les protocoles (source de l'interaction, destination de l'interaction, propagation de l'interaction). La quatrième étape définit le tableau d'interaction des fonctions et donne la distribution des interactions entre les fonctions de chaque protocole considéré l'un après l'autre. La cinquième étape de la méthode RCL permet de déduire les modèles d'interaction des AACL classées par catégorie et donne un aperçu de la complexité des échanges créés par ces interactions cross–layer. La sixième étape produit les tableaux de description des interactions par protocole qui décrivent l'usage concret fait de chaque AACL par les fonctions des différents protocoles. L'étape six de la méthode RCL donne un aperçu du travail de conversion à réaliser et de sa complexité. Elle constitue l'amorce de la transition entre l'aspect conceptuel et théorique de l'étude des AACL et la phase pratique de mise en œuvre qui doit déboucher sur la mesure des gains de performance obtenus. La septième étape de la méthode standardise l'implantation des AACL. Elle permet de prendre en compte les mécanismes de communication du modèle global qui ont été déjà arrêtés.

Mécanisme cross–layer de temporisation persistante :

Sur la base de la proposition de mécanismes novateurs réalisée à l'étape six de l'application de la méthode RCL, notamment sur le protocole fiable de la couche transport choisi, nous avons orienté notre étude sur l'impact de l'état du canal sur ces protocoles. La nature variable et imprévisible de l'état du canal sans fil est la principale différence qui existe entre l'environnement filaire et l'environnement sans fil. Elle constitue le handicap majeur pour lequel l'amélioration des performances des réseaux sans fil devient une nécessité au fur et à mesure de leur expansion. Ainsi, pour poursuivre l'adaptation du protocole TCP à l'environnement sans fil, nous avons proposé une politique de temporisation qui utilise les informations d'état du canal fournies par la couche liaison à travers le sous–système environnement du modèle conceptuel cross–layer. La politique de temporisation actuelle de TCP, lorsque l'état du canal est mauvais, se traduit par l'envoi de segments TCP à chaque expiration du temporisateur d'attente qui évolue selon un algorithme de back–off exponentiel, et surtout, ce mécanisme se traduit par l'attente de l'expiration de ce temporisateur avant l'envoi d'un message dans le réseau sans fil même si l'état du canal redevient favorable. C'est pourquoi, la politique persistante que nous proposons consiste à observer le prochain changement favorable de l'état du canal pour envoyer un paquet. Ses avantages en terme de latence, de débit et de taux de tentatives infructueuses ainsi que de consommation d'énergie, ont été démontrés dans les modèles mathématiques et traduits par les résultats des simulations.

A la différence de TCP, le protocole SCTP intègre un mécanisme de gestion d'association auquel est associé le mécanisme de détection et de gestion de la perte de chemin dans le réseau. La détection et la gestion de la perte de chemin utilise des messages de signalisation périodiques. Les similitudes entre les deux protocoles se rapportent entre autres aux trois phases de fonctionnement : démarrage lent, évitement de congestion et recouvrement rapide. Le recours au back–off exponentiel pour les retransmissions de messages constituent également un autre point commun entre les deux protocoles. La démarche conceptuelle étant constructive pour toutes les raisons évoquées, notamment relatives à la conservation des avantages de l'architecture, nous avons procédé à l'analyse de l'impact de l'utilisation du protocole SCTP en remplacement de TCP en appliquant la méthode RCL. Les modèles conceptuels cross–layer produits nous amènent à la conclusion de la convergence de réaction

des deux protocoles aux interactions cross–layer identifiées dans la deuxième étape de la méthode. C'est pourquoi, nous avons étendu l'application de la temporisation persistante au protocole SCTP lorsqu'il est utilisé dans un canal à état variable. L'évaluation de performances de l'utilisation d'un autre protocole transport fiable a conforté l'avantage de l'utilisation de la temporisation persistante qui reste une solution efficace en remplacement du back–off exponentiel utilisé par le mécanisme de retransmission des deux protocoles, dans le contexte d'un mauvais état du canal. Les résultats obtenus ont montré l'avantage de la temporisation persistante lorsque le nœud mobile est soumis aux aléas de la variation dynamique de l'état du canal, en terme de latence, de débit et de consommation d'énergie du fait des tentatives infructueuses de retransmission. La comparaison croisée des résultats obtenus pour chacun des protocoles a reflété l'efficacité des mécanismes des deux protocoles TCP et SCTP en raison de leur différence relative de fonctionnement.

Mécanisme cross–layer d'évaluation continue de l'état du canal :

La philosophie de base de la temporisation persistante repose sur l'exploitation de l'état du canal sans fil dont la caractéristique essentielle est que sa qualité varie en fonction de l'espace et du temps. L'information sur l'état du canal est fournie par la couche liaison. Pour cela, elle utilise les paramètres du sous–système environnement et les modules de notification explicite qu'il comporte. L'évaluation de l'état du canal est tributaire de l'activité du réseau, qu'elle soit une activité interne ou externe au nœud sans fil. Nous avons proposé le mécanisme cross–layer de temporisation persistante au niveau des protocoles de transport fiable lors d'un mauvais état du canal pour optimiser l'utilisation non efficace du back–off exponentiel. Le comportement persistant a la particularité d'engendrer la suspension de l'activité de retransmission au niveau de la couche transport, mais nécessite une continuité de l'évaluation de l'état du canal. Pour compléter ce modèle cross–layer de temporisation, nous avons proposé le mécanisme d'évaluation continue de l'état du canal. Ce dernier mécanisme cross–layer repose sur la distinction des cas possibles qui peuvent survenir. L'activité ambiante et l'activité périodique des protocoles de routage pro–actif ont d'abord été prises en compte. Le mécanisme complémentaire est déclenché lorsque les paramètres d'état du canal n'ont pas subi de mise à jour pendant un temps déterminé. Dans ce cas, le système réagit en sollicitant le protocole de routage pour qu'il procède à la mise à jour de route pour les paquets en cours de transmission. Ce mécanisme permet ainsi de poursuivre l'évaluation de l'état du canal, qui est nécessaire au bon déroulement de la temporisation persistante.

Standardisation des données cross–layer fournies par les protocoles de routage :

Les protocoles de routage apportent une contribution non négligeable aux modèles et mécanismes cross–layer. Par exemple, les protocoles pro–actifs facilitent l'évaluation continue de l'état du canal lorsque la temporisation persistante est mise en œuvre par la couche transport en réponse à un mauvais état du canal. Pour rendre les modèles cross–layer fonctionnels indépendants du protocole de routage utilisé, nous avons procédé à l'application des trois premières étapes de la méthode RCL aux piles de protocoles dont le routage est assuré par un protocole choisi dans la famille des pro–actifs (OLSR et DSDV) et celle des ré–actifs (AODV et DSR). Cette étude a permis d'enrichir les modèles cross–layer avec des AACL de mise à disposition des données provenant des structures de routage et des AACL de notification, notamment de gigue d'envoi de paquets. De ce fait, il a été possible de faire ressortir les points de convergence et de différence entre ces protocoles. A partir de ces points, nous avons défini une structure standard composée de trois tables destinées à rendre transparent le changement du protocole de routage dans le fonctionnement des modèles cross–layer. La table des voisins à 1 saut, celle des nœuds accessibles à plus d'un saut et celle des nœuds inaccessibles constituent le premier axe des propositions que nous avons faites pour renforcer le fonctionnement des modèles et mécanismes cross–layer qui ont été établis.

L'étude comparative des AACL générées par les protocoles de routage et la considération de la problématique de transmission des paquets perçue au niveau des protocoles transport fiables, nous ont conduit à la proposition de la table LAST (Link Access State Table). L'objectif est d'indiquer l'état de l'acheminement d'un message émis par la couche transport. Un message peut être soit en cours de transmission, soit en attente, soit en état d'abandon de sa transmission. En exploitant les données de cette table, la couche transport peut avoir recours à la temporisation persistante lorsque des messages de changement de route dû à la mobilité des nœuds parviennent au nœud émetteur.

Mécanisme cross–layer de temporisation unifiée :

Les protocoles de routage ré–actifs observent une temporisation lors de la transmission des messages qu'ils reçoivent de la couche transport, tout comme le mécanisme d'émission des paquets des protocoles de transport fiable. Nous avons procédé à l'étude des primitives d'émission de messages de ces protocoles dans le cas où un protocole fiable est utilisé en combinaison avec un protocole de routage ré–actif. Le mécanisme cross–layer de temporisation unifiée que nous avons proposé est destiné à minimiser la consommation d'énergie par répétition du nombre de tentatives infructueuses au niveau des deux couches. Cette proposition vise à économiser la consommation d'énergie dans ce cas de duplication de la temporisation d'attente d'envoi, à des échelles de temps différentes, entre la couche transport et la couche réseau. Etant donné les raisons qui expliquent ces retransmissions, nous avons accompagné la proposition de ce mécanisme cross–layer de temporisation unifiée par un évitement de retour à la phase de démarrage lent préjudiciable au débit de transmission des protocoles de transport fiable. La prise en compte de ce débit fait que la temporisation unifiée est également accompagnée de l'adoption possible par les protocoles transports fiables d'un comportement Vegas. Ainsi, la valeur de la fenêtre de congestion, qui détermine le débit à transmettre, peut prendre des valeurs stagnantes ou évolutives, en fonction du temps de réponse du réseau.

Perspectives :

Les différentes propositions que nous avons faites dans cette thèse ouvrent un certain nombre de perspectives :

- Les modèles et mécanismes cross–layer ne sont pas finis, mais sont divers et variés. Ils dépendent à la fois des protocoles utilisés dans la pile, des services à partager, des paramètres significatifs à exporter ou à utiliser et des événements qui surviennent lors de l'exécution d'un protocole. C'est pourquoi nous considérons la méthode RCL comme évolutive, en ce sens qu'elle peut être améliorée au besoin, pour prendre en compte des éléments supplémentaires permettant de juger de l'impact des interactions cross–layer et de la complexité des modifications à apporter au système initial.

- Une autre approche comparative de la politique persistante de temporisation de TCP avec la politique traditionnelle se fait en intégrant les paramètres d'amélioration de débit dont l'évitement du recours au démarrage lent lors de la reprise du trafic après un blocage temporaire des envois dû à un état défavorable du canal. Ce principe permet d'améliorer les gains obtenus dans les résultats précédents.

- En ce qui concerne la standardisation des données cross–layer fournies par les protocoles de routage, pour les trois tables proposées et la table LAST, nous envisageons une évaluation des apports de cette standardisation pour confirmer les gains de performances intuitifs sous–jacents à la proposition.

- Pour le mécanisme cross–layer de temporisation unifiée, la quantification des apports de la temporisation unifiée reste une perspective ouverte.

Le travail que nous avons effectué est encourageant et donne un aperçu global de la complexité du domaine cross–layer. Les techniques cross–layer sont émergentes et constituent un domaine qui recouvre beaucoup de travail à réaliser, chaque cas particulier d'interaction cross–layer doit faire l'objet d'une étude spécifique.

Bibliographie

[AGH06] H. Badis, K. Al Agha, "**Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR)**", Internet Draft, March 2006.

[BAL98] H. Balakrishnan and R. H. Katz, "**Explicit Loss Notification and Wireless Web Performance**", Proc. IEEE GLOBECOM Global Interne, Sydney, Australia, Nov 1998. http://nms.lcs.mit.edu/papers/

[BAL99] K. Balachandran, S. R. Kadaba, S. Nanda, "**Channel quality estimation and rate adaptation for cellular mobile radio**", IEEE Journal on Selected Areas in Communications 1999 ; 17(7) :1244-1256.

[BHU04] V. Bhuvaneshwar, M. Krunz, A. Muqattash, "**CONSET : A Cross–layer Power Aware Protocol for Mobile Ad–hoc Networks",** Communications, 2004 IEEE International Conference on , Volume: 7 , 20-24 June 2004, Pages:4067 - 4071 Vol.7.

[BRA01] B. E. Braswell, J. C. McEachen, "**Modelling data rate agility in the IEEE 802.11a WLAN Protocol**", in OPNETWORK 2001, March 2001.

[BRO98] J. Broch, D. Maltz, D. B. Johnson, Y.C. Hu and J. Jetcheva, "**A performance comparison of Multi-Hop Wireless Ad–hoc Network Routing Protocols.**" In Proc. ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom98), pages 85 97, 1998.

[CON04] M. Conti, G. Maselli, G. Turi, S. Giordano, **"Cross–layering in Mobile Ad–hoc Network Design"**, IEEE Computer Magazine, Volume 37, Issue 2, February 2004 Page(s) : 48 – 51.

[COR01] S. Corson, P. Vincent, "**Temporally Ordered Routing Algorithm (TORA)**", Internet – Draft, IETF MANET Working Group, August 2001.

[DAS01] S. Das, C. Perkins, E. Royer, "**Ad–hoc On Demand Distance Vector Routing (AODV)**", Internet – Draft, IETF MANET Working Group, November 2001.

[DEM01] T. Demir, "**Simulation of Ad–hoc Networks with DSR Protocol**", May 2001, http://netlab.boun.edu.tr/papers/Iscis2001-DSR-TamerDEMIR+.pdf.

[DIA03] M. Diaz, "**Cours de multimédia**", formation DEA RT INP-ENSEEIHT, jan 2003.

[DOS02] S. Doshi, S. Bhandare, and T. X. Brown "**An on-demand minimum energy routing protocol for a wireless ad–hoc network**", In ACM SIGMOBILE Mobile Computing and Communications Review, volume 6, pages 50-66, july 2002.

[DSR02] " **The Dynamic Source Routing Protocol for Mobile Ad–hoc Networks (DSR)**", 21 february 2002, http://www.ietf.org/proceedings/02mar/I-D/draft-ietf-manet-dsr-07.txt.

[FAL03] K. Fall and K. Varadhan. "**The ns manual**", the VINT Project. http://www.isi.edu.nsnam/ns/doc/index.html, December 2003.

[GAL98] R. G. Gallager, "**Energy limited channels : coding, Multi – access, and Spread Spectrum**", 1998 Conf. Info. Sci Sys. March 1998.

[GER01] M. Gerla, X. Hong, "**Alternative OSPF ABR Implementations**", Internet – Draft, IETF MANET Working Group, December 2001.

[GLO] "**GloMoSim : A Scalable Network Simulation Environment**", http://pcl.cs.ucla.edu/projects/glomosim/

[HAA98] Z. J. Haas, M. R. Pearlman, "**The Zone Routing Protocol (ZRP) for Ad Hoc Networks**", INTERNET-DRAFT, August 1998.

[HAR05] I. Haratcherev, J.Taal, K. Langendoen,R. Lagendijk and H. Sips, "**Automatic IEEE 802.11 rate control for streaming applications**", Wireless Communications and Mobile Computing 2005; 5:421-437.

[HAV06] H. Haverinen, Ed., J. Salowey, "**Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)**", RFC 4186, Ed. January 2006.

[IEE03] "**IEEE standard 802.11h supplement. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications**". Spectrum and transmit power management extensions in the 5 Ghz band in Europe, 2003.

[JAC01] P. Jacquet, T. Clausen, "**Optimized Link State Routing Protocol (OLSR)**", Internet – Draft, IETF MANET Working Group, October 2001.

[JET01] J. G. Jetcheva, Y. Hu, D. Johnson, D. Maltz, "**The Dynamic Source Routing Protocol for Mobile Ad–hoc Networks (DSR)**", Internet Draft, IETF MANET working group, Nov 2001.

[JIA99] M. Jiang, J. Li, Y.C. Tay, "**Cluster Based Routing Protocol(CBRP)**", INTERNET-DRAFT, July 1999.

[JOH98] D. B. Johnson, D. A. Maltz, "**Dynamic Source Routing in Ad–hoc Wireless Networks**", Mars 1998, http://www.monarch.cs.cmu..edu/monarch -papers/.

[KAM97] A. Kamerman, L. Monteban, "**WaveLAN II : a high performance wireless LAN for the unlicensed band**", Bell Labs Technical Journal, 1997 ; Summer : 118-133.

[KAW05] V. Kawadia, P.R. Kumar, "**A cautionary perspective on cross-layer design**", IEEE Wireless Communications, Volume 12, Issue 1, Feb. 2005, Page(s):3 – 11.

[KHA03] S. A. Khayam, S. Karande, M. Krappel, H. Radha, **"Cross-Layer Protocol design for real-time multimedia applications over 802.11b networks",** Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on , Volume: 2 , 6-9 July 2003, Pages:II - 425-8 vol.2.

[KRI97] K.J. Krizman, T.E. Biedka, and T. S. Rappaport. "**Wireless position location : Fundamentals implementation strategies, and source of error**", In Proceedings of the IEEE Vehicular Tech. Conference, volume 2, pages 919 – 923, 1997.

[LAO98] A. Laouti, P. Muhlethaler, A. Najid, E. Plakoo, "**Simulation Results of the OLSR Routing Protocol for Wireless Network**", INRIA Rocquencourt, April 1998. http://menetou.inria.fr/~muhletha/medhoc.pdf

[LAR99a] L. Larzon, M. Degemark and S. Pink, "**Efficient use of Wireless Bandwidth for multimedia Applications**", IEEE International Workshop on Mobile Multimedia Communications (MoMUC), 15-17 November 1999, Page(s):187 – 19.

[LAR99b] L. Larzon , M. Degemark and S. Pink, "**UDP Lite for Real Time Multimedia Applications**", IEEE International Conference of Communications (ICC), June 1999.

[LI03] W. Li, Z. Bao – yu, "**Study on Cross–layer Design and Power Conservation in Ad–hoc Network**", IEEE PDCAT'2003, 27-29 Aug. 2003 Pages:324 – 328.

[MAL02] M. Maleki, K. Dantu, and M. Pedram, "**Power-aware source routing protocol for mobile ad–hoc networks**", In proceedings of the ACM International Symposium on Low Power Electronics and Design, pages 72 –-5, August 2002.

[MAL94] L. Brakmo, S. O'Malley, "**TCP Vegas : New Techniques fonr Congestion Detection and Avoidance**", in SIGCOMM'94 Conference on Communications, Architectures and Protocols, (London, United Kingdom), pp.24-35, October 1994.

[MAN] "**Mobile Ad–hoc Network (MANET)**", http://www.ietf.org/

[MAZ88] C. Mazel, "**évaluation des performances par simulations – application aux canaux de signalisation de systèmes radio téléphoniques**", thèse de doctorat INP de Grenoble, P149 – 156, Juin 1988.

[MIN02] Y. Min-hua, L. Yu, Z. Hui-min, "**The IP Handoff between Hybrid Networks**", Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on , Volume: 1 , 15-18 Sept. 2002, Pages:265 - 269 vol.1.

[MUQ03] A. Muqattash and M. Krunz. "**Power controlled dual channel (PCDC) medium access protocol for wireless ad–hoc networks**", In proceedings of the IEEE INFOCOM Conference, volume 1, pages 470 – 480, April 2003.

[NI02] Q. Ni, L. Romdhani, T. Turletti, and I. Aad, "**QoS Issues and Enhancements for IEEE 802.11 Wireless LAN**", Inria Sophia – Antipolis, RR 4612, November 2002.

[NI99] S.-Y. Ni, Y.-C. Tseng, Y. –S. Chen, and J.-P. Sheu. "**The broadcast storm problem in a mobile ad–hoc network**", In Proceedings of the ACM MobiCOm Conference, pages 151-162, 1999.

[PAV03] J. D. P. Pavon, S. Choi, "**Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement**", In IEEE International Conference on Communications, 2003 (ICC' 03), Vol 2, Anchorage, Alaska, USA, May 2003; pp 1108-1113.

[PEI00] G. Pei, M. Gerla, X. Hong, "**LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility**", Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing, Boston, Massachusetts, Pages: 11 – 18, 2000.

[PER01a] C. Perkins, P. Bhagwat, "**Destination Sequenced Distance Vector (DSDV)**", Internet – Draft, IETF MANET Working Group, November 2001.

[PER01b] R. Samir Das, C. E. Perkins and Elizabeth M. Royer, "**Performance Comparison of Two On-demand Routing Protocols for Ad–hoc Networks**", In IEEE Personal Communications, vol 8, no1, pp. 16 28, February 2001.

[RAI02] V. T. Raisinghani, A. K. Singh, S. Iyer, **"Improving TCP performance over Mobile Wireless Environments using Cross–layer Feedback"**, Personal Wireless Communications, 2002 IEEE International Conference on , 15-17 Dec. 2002, p81–85.

[RAM01] K. Ramakrishnan, S. Floyd, D. Black. "**The Addition of Explicit Congestion Notification (ECN) to IP**", RFC3168 September 2001, www.ietf.org.

[RFC0768] J. Postel , "**User Datagram Protocol**", STD 6, RFC 768, August 1980.

[RFC0791] J. Postel. "**Internet Protocol**", RFC 0791, Sep-01-1981, www.ietf.org.

[RFC0793] J. Postel. "**Transmission Control Protocol**", RFC 0793 Sep-01-1981, www.ietf.org.

[RFC1122] R. Braden, "**Requirements for Internet Hosts – Communication Layers**", RFC 1122, October 1989.

[RFC1349] P. Almquist "**Type of Service in the Internet Protocol Suite**", RFC 1349, July 1992, www.ietf.org.

[RFC1812] F. Baker, "**Requirements for IP Version 4 Routers**", RFC1812, Ed. June 1995.

[RFC2018] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, "**TCP Selective Acknowledgement Options**", RFC 2018, October 1996.

[RFC2330] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, "**Framework for IP performance metrics**", RFC 2330, Internet Engineering Task Force, May 1998.

[RFC2501] "**Mobile Ad–hoc Networking (MANET) : Routing Protocol Performance Issues and Evaluation Considerations**", RFC 2501, S. Corson, J. Macker, Janvier 1999.

[RFC2581] M. Allman, V. Paxon, W. Stevens, "**TCP Congestion Control**", RFC 2581, April 1999.

[RFC2582] S. Floyd, T. Henderson, "**The NewReno Modification to TCP's Fast Recovery Algorithm**", RFC 2582, April 1999.

[RFC2883] S. Floyd, J. Mahdavi, M. Mathis, M. Podlosky, "**An Extension to the Selective Acknowledgement (SACK) Option for TCP**", RFC 2883, August 1999.

[RFC2960] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson, "**Stream Control Transmission Protocol**", RFC 2960, October 2000.

[RFC2998] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, E. Felstaine, "**A framework for Integrated Services Operation over DiffServ Networks**", RFC 2998, November 2000.

[RFC3155] S. Dawkins, G. Montenegro, M. Kojo, V. Magret, N. Vaida, "**End-to-End Performance Implications of Links with Errors**", RFC 3155, August 2001.

[RFC3366] G. Fairhurst, L. Wood, "**Advice to link designers on link Automatic Repeat reQuest (ARQ)**", RFC 3366, August 2002.

[RFC3453] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, J. Crowcroft, "**The use of Forward Error Correction (FEC) in Reliable Multicast**", RFC 3453, December 2002.

[RFC3684] R. Ogier, F. Templin, M. Lewis, "**Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)**" RFC 3684, February 2004.

[ROY00] E. M. Royer, C. E. Perkins, "**Multicast Operation of the Ad–hoc On Demand Distance Vector Routing Protocol**", Internet – Draft, IETF MANET Working Group, August 2000.

[ROY99] E. M. Royer, C. E. Perkins, "**Ad–hoc On Demand Distance Vector Routing (AODV)**", second annual IEEE workshop on Mobile Computing Systems and Applications, February 1999, pp. 90 – 100.

[SAM98] T. Ue, S. Sampei, N. Morinaga, K. Hamaguchi, "**Symbol rate and modulation level controlled adaptive modulation/TDAM/TDD system for high bit rate wireless data transmission**", IEEE Transactions on Vehicular Technology 1998 ; 47(4) : 1134-1147.

[SCH05] M. V. D. Schaar, N. S. Shankar, **"Cross–layer Wireless Multimédia Transmission : Challenges, Principles and New Paradigms"**, Wireless Communications, IEEE [see also IEEE Personal Communications], Volume 12, Issue 4, Aug. 2005 Page(s):50 – 58.

[SHA03] S. Shakkottai and T. S. Rappaport, P. C. Karlsson, **"Cross-Layer Design for Wireless Network"**, IEEE Communications Magazine, Volume 41, Issue 10, October 2003 Page(s) :74 – 80.

[TOR97] "**A performance comparison of TORA and ideal Link State Routing**", february 1997,
http://tonnant.itd.nrl.navy.mil/tora/tora_sim.html

[VEG02] A. J. Van DerVegt, "**Auto rate fallback algorithm for the IEEE 802.11a Standard**", Technical report, Utrecht University, 2002.

[VER01] A. Veres, A.T. Campbell, M. Barry, and L.H. Sun, "**Supporting service differenciation in wireless packet networks using distributed control**", IEEE journal of Selected Areas in Communications (JSAC), Special Issue on Mobility and Ressource Management in Next – Generation Wireless Systems, Vol. 19, No 10, pp. 2094-2104, October 2001.

[WAN03] Q. Wang, M. A. Abu-Rgheff, **"Cross–layer Signalling for Next – Generation Wireless Systems",** IEEE WCNC 2003, Volume: 2 , 16-20 Mar 2003 p1084-1089 vol.2.

[YU02] W. Yu and J. Lee. "**DSR-based energy-aware routing protocols in ad–hoc networks**", In Proceedings of the International Conference on Wireless Networks, June 2002.

[ZHE03] H. Zheng, **"Optimising Wireless Multimedia Transmissions Through Cross–layer Design",** Multimedia and Expo, 2003. ICME '03. Proceedings. Volume 1, 6-9 July 2003 Page(s):I – 185 –8 vol.1.

[ZIN01] A. Zinin, "**Alternative OSPF ABR Implementations**", Internet – Draft, IETF MANET Working Group, February 2001.

Publications

Communications dans les conférences internationales :

[1] M. I. Tiado, R. Dhaou, A-L. Beylot. "**Multilevel Network Modelling to Achieve Cross–layer Mechanisms**", Med-Hoc-Net'05 Conference, Île de Porquerolles, France, IFIP Serie, pp79–89, Springer 2006.

[2] M. I. Tiado, R. Dhaou, A-L. Beylot. "**RCL: A new method for Cross-Layer Network Modelling and Simulation**", MWCN Conference, Marakech Maroco, September 2005, 19-21th.

[3] R.Dhaou, V. Gauthier, M. Issoufou Tiado, M. Becker, A.-L. Beylot, "**Cross–Layer Simulation: Application to Performance Modelling of Networks composed of MANETs and Satellites**", Tutorial, In: 2nd International Conference on Performance Modelling and Evaluation of Heterogeneous Networks, HET'NET'04, Ilkley, Angleterre, Juillet 2004, pp. T11.1-T11.30.

Article Soumis :

[4] M. Issoufou Tiado, R. Dhaou, A.-L. Beylot, "**Persistent TCP Timeout Policy by using Cross–Layer Mechanism**", In: IEEE GLOBECOM Technical Conference & IEEE COMMUNICATIONS EXPO, Submitted to IEEE GlobeCOM'06, 27 November – 1st December 2006, San Francisco, California, USA.

Rapport de Recherche :

[5] M. I. Tiado, R. Dhaou, A. L. Beylot. "**UCL: A new method for Cross-Layer Network Modelling**", Research communication, IRIT/2005-1-R, IRIT, ENSEEIHT, Toulouse., jan 2005. http://www.enseeiht.fr/~beylot/IRITBeylot5.pdf
<http://www.enseeiht.fr/%7Ebeylot/IRITBeylot5.pdf>